REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	31 March 1998	Technical Report

4. TITLE AND SUBTITLE
Hybrid System Extraction of Control Automata with Small Topologies

5. FUNDING NUMBERS
DAAH04-96-1-0341

6. AUTHOR(S)
A. Nerode, J. B. Remmel and A. Yakhnis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Regents of the University of California
c/o Sponsored Projects Office
336 Sproul Hall
Berkeley, CA 94720-5940

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 35893.81-MA-mur

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by the documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
We propose a game framework for analyzing, extracting and verifying digital control programs for continuous plants by regarding such programs as finite state winning strategies in associated games. We call such interacting systems of digital control protrams and continuous plants "hybrid systems" and model them as networks of interacting concurrent digital programs or automata. this extends to hybrid systems the paradigm introduced by Nerode, Yakhnis and Yakhnis for analysing concurrent digital programs meeting program specifications as winning finite state strategies in associated two person games

14. SUBJECT TERMS
control strategies, hybrid systems, continuous sensing control automata

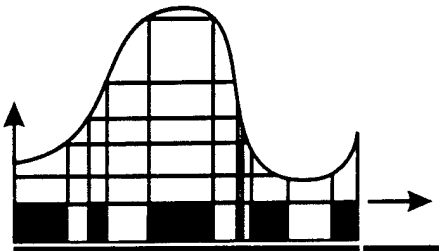
15. NUMBER OF PAGES
43

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev.2-89)
Prescribed by ANSI Std. 239-18
298-102



Center for Foundations of Intelligent Systems

Technical Report
97-03

Hybrid System Games: Extraction of Control Automata with Small Topologies

A. Nerode, J. B. Remmel and A. Yakhnis

June 1997

CORNELL
UNIVERSITY

625 Rhodes Hall, Ithaca, NY 14853 (607) 255-8005

Technical Report
97-03

Hybrid System Games: Extraction of Control Automata with Small Topologies

A. Nerode, J. B. Remmel and A. Yakhnis

June 1997

Hybrid System Games: Extraction of Control Automata with Small Topologies

Anil Nerode*[1] and Jeffrey B. Remmel**[2] and Alexander Yakhnis***

[1] Mathematical Sciences Institute
Cornell University, Ithaca, NY 14853
e-mail: anil@math.cornell.edu
[2] HyBrithms Corp. HyBrithms Corp.†
11201 S.E. 8th Street, Bldg. J, Suite 140
Bellevue, Washington and
Department of Mathematics
University of California at San Diego San Diego, CA 92093
e-mail; jremmel@ucsd.edu
[3] Mathematical Sciences Institute
Cornell University, Ithaca, NY 14853
e-mail: ayakh@math.cornell.edu

1 Introduction

Hybrid control is the control of continuous plants by sequential automata. This usually means frequent changes in the continuous conventional control law applied to the plant, changes based on sensor measurements of the trajectory. This typically yields plant trajectories without smooth tangents at the discrete times when the control law ordered by the control program changes. How and when to make these control law changes is the business of the sequential automaton. The question is then how should we model this and how can we find control sequential automata to meet a prescribed performance specification.

We propose a game framework for analyzing, extracting and verifying digital control programs for continuous plants by regarding such programs as finite state winning strategies in associated games. We call such interacting systems of digital control programs and continuous plants "hybrid systems" and model them as networks of interacting concurrent digital programs or automata, following [36], [37]. This extends to hybrid systems the paradigm introduced by A. Nerode, A. Yakhnis, and V. Yakhnis [38] for analyzing concurrent digital programs meeting program specifications as winning finite state strategies in associated two person games. This hybrid game formulation is intended to facilitate the transfer of recent tools from logic, concurrency, and dynamical systems to extraction and verification of digital control programs for continuous systems. Hybrid Games also facilitate infusion into

* Research supported by ARO under the MURI program "Integrated Approach to Intelligent Systems", grant no. DAAHO4-96-1-0341.
** Research supported by Dept. of Commerce Agreement 70-NANB5H1164.
*** supported by DARPA- US ARMY AMCCOM (Picatinny Arsenal, N. J.) contract DAAA21-92-C-0013 to ORA Corp.
† HyBrithms Corp was formerly know as Sagent Corporation

hybrid systems theory of many ideas from the traditional differential game approach to control.

The Basic Model

We now introduce our basic model for Hybrid Control which is essentially the same as the model discussed in [24]. A finite control automaton is an automaton with finite input and output alphabets and a finite number of internal states. Its input letters are fired by measurements of plant state. Its output letters are control signals, that is mode switches, for the plant controller. Our basic model for a simple hybrid system consists of the following.

1. A finite control automaton, which is usually thought of as some sort of logical device or program which makes inferences based on current information about the plant state to deduce when to change control laws for the plant. See Kohn-Nerode [24], [25].

2. A continuous plant controller obeying the control law currently supplied by the finite control automaton.

3. A continuous plant being controlled. We include in the plant the physical plant controller (actuator), but not the finite control automaton feeding control orders (mode switches) to the physical plant controller.

4. An analog-to-digital or, equivalently, a signal to symbol, converter supplying to the finite control automaton as input digitized sensor data sampled from the plant.

5. A digital-to-analog or, equivalently, a symbol to signal converter converting symbolic control orders output by the control automaton into a control function of time regulating the parameters of the physical plant controller.

These elements are pictured in Figure 1.

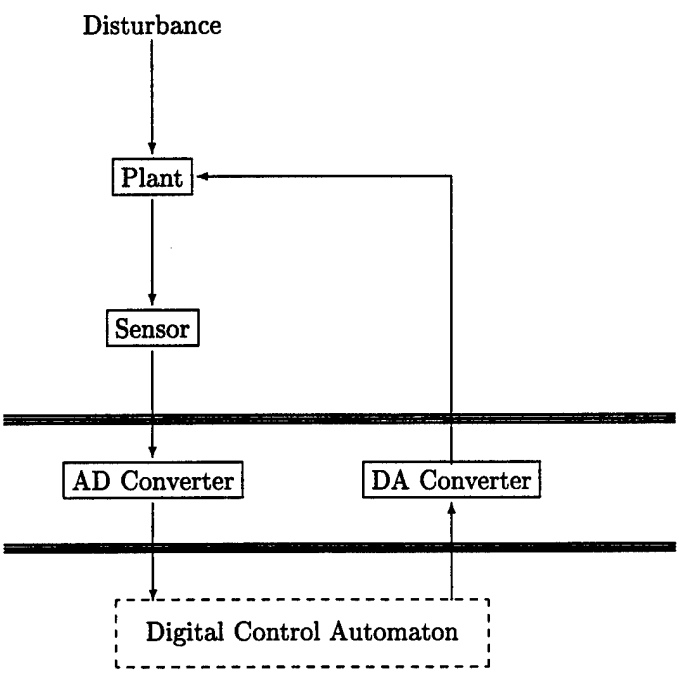


Figure 1.

We summarize the essential features of hybrid systems model of [24]. We think of the sequential control automaton as completing "work cycles" in successive intervals Δt_k of time. During the interval Δt_k, a control law u_k which is imposed by the sequential control automaton at the end of the previous interval is active in controlling the system. Also the sequential control automaton is subject during Δt_k to an input v_k to the system. During the first phase of Δt_k, the sequential control automaton is accumulating a sensor data history s about the system through the analog-to-digital converter. The sequential control automaton starts interval Δt_k in a certain initial state, uses s to compute a new control law u_{k+1} and a new automaton state and, at the end of Δt_k, it outputs u_{k+1} through the digital-to-analog converter to the plant controller for use in the next interval Δt_{k+1}. Then all processes start over for Δt_{k+1}. We envisage the input as encoding all the partial information available to the control automaton about the state of the plant. A hybrid control run thus will be a possibly infinite sequence

$$u_0, v_0, u_1, v_1, \ldots.$$

We shall see that it is very natural to view such a hybrid control run as play of a game between two players, Plant and Control. That is, Control and Plant alternate moves in a game in which Control moves by listing full information about control law

u_k for Plant's use, and then Plant moves by listing the partial information v_k about the plant state for Control's use. The range of values of u_k and v_k and the relationship between u_k and v_k is dependent on the particular application. Then, in the spirit of [38], we can view a successful sequential control automaton as implementing a winning strategy for Control. That is, in any play in which Control follows the winning strategy and Plant plays according to the rules of the game, i.e. follows its differential equations, the plant trajectories will meet the desired performance specification.

Performance Specifications

Our performance specifications are usually open sets of trajectories. Quoting an example of Kohn, the Boeing 737 was to be designed so that if a cup of coffee is no more than 3/4 full anywhere in the aircraft, it never spills during maneuvers. This is not a conventional optimality requirement. It is a "perform sufficiently well" criterion which we call an ϵ-performance criterion or, alternately, an ϵ-optimality condition. For example, such a criterion might require that we produce a trajectory whose costs is within a user defined ϵ of the minimum cost trajectory.

Outline of Paper

In sections 2 and 3, we present game models for extraction, analysis, and verification of control strategies for simple hybrid systems. All games will be between two players, Plant and Control. The objective of any game is for Control to force Plant to obey its performance specification. In the game model presented in section 2, measurements of Plant state are made at discrete times (discrete sensing) and changes in the control order to the plant are also made at discrete times (discrete mode switching). Such games are an adaptation to hybrid systems of the games of A. Nerode, A. Yahknis, and V. Yahkinis [38]. The latter were introduced to extract, analyze, and verify digital concurrent programs.

In Section 3, we introduce "continuous sensing games" to model plants characterized by continuous dynamics, such as a system of ordinary differential equations, with a controller which continuously senses the plant state. We assume that the controller is allowed to reset the parameters of the plant dynamics at a sequence of discrete times only. Such controllers allow us to model directly analog sensors which continuously sense the plant state and which output exact real number control parameters to the physical plant controller at discrete times. In most cases, we will assume that the values output by the controller are purely digital, one of a finite number of control order, to be implemented by a digital to analog converter. Even in such cases, such a controller must be regarded as non-digital if the input values allowed are exact real numbers, even if there are only a finite number of internal states and a finite number of output control orders (mode switches). Our methodology is to start by extracting a continuously sensing, discretely acting, feedback control function which meets the performance specification. The discrete sensing games of Section 3 can then be thought of as subgames of continuous sensing games in which the information sensed between discrete sampling times is ignored.

Thus continuous sensing games are a second class of games between two players, Plant and Control. After each control order is sent by Control to the plant, Plant displays a segment of a plant trajectory y which begins when control order is given, with initial condition the plant state at that time, and which ends when the next control order is issued. We can think of this segment as a contiguous block of Plant

moves, one at each time in that interval. In this picture the Plant move at a time $\tau > t$ in the interval is just the plant state $y(\tau)$. In the same picture, a Control move occurs at the same instant τ and is either *no action* or a control order. We call the latter moves essential Control moves. We assume that essential Control moves occur only at a discrete sequence of times. Each such time is the end of a block of Plant moves. According to this picture, in a continuous sensing game, Plant plays continuously, Control has continuous knowledge of Plant moves, and Control makes essential moves only once in a while.

The motivation for introducing continuous sensing games is that they help us extract strategies for a digital controller which will meet performance specifications. The idea is that it often easier to find a (non-digital) continuous state strategy for Control in a continuous sensing game which forces the plant to meet performance specifications. We then extract a finite state strategy for a finite state digital controller doing approximately the same thing by approximating to the continuous strategy for the non-digital controller using the Kohn-Nerode method of extracting finite control automata from finite open covers.

In Section 4, we discuss performance specifications and the Kohn-Nerode method cover method ([24], appendix 2). It works as follows. Suppose we are given a controller which meet an open specification. Then the Kohn-Nerode method takes an open finite cover of that controller within the open specification and interprets it as:

1. A finite automaton with a finite input alphabet and a finite state alphabet.
2. A digital to analog converter.
3. An analog to digital converter.
4. A control automaton for the plant.

When considered as a hybrid system, the plant plus the automaton derived from the cover forces the plant to obey the open performance specification. We carry out this process for a simple model of a water pump used to maintain a certain range of values of the water in a water tank. That is, we shall explicitly construct a strategy for Control in a continuous sensing game which models this system and then show how we can easily approximate such a strategy to construct a strategy for Control in a corresponding discrete sensing game. Finally we shall show how this enables us to design a digital control automaton for the hybrid system which meets the performance specifications as well as how to construct the Kohn-Nerode small topologies for the hybrid system which will verify the controllability-observability of the system in the sense of [24].

2 Games with Discrete Sensing and Discrete Mode Switching

In this section, we provide a game setting for the specification, extraction, and verification of digital control programs for hybrid systems. Extracting a control program for a continuous plant which forces the plant to obey a performance specification is identified with extracting a winning finite state strategy in an associated game. The performance specification itself is identified with a set of acceptable plant state

trajectories. The games introduced in this section and the next section, each have two players, Control and Plant. In our games, we represent the effect on plant state of unknown disturbances and uncertain measurements by allowing multiple legal moves for Plant. For example, one source of multiple possible moves for Plant is that, with a given initial condition, each disturbance over a time interval $[t, t + \Delta]$ can yield a different plant state trajectory over that interval and hence a different final plant state at the end of the time interval. Another source of multiple Plant moves is measurement errors. We assume Control sends perfect information to the physical plant controller, namely a suitable control law for the next interval of time. However Plant sends imperfect information to the Control program, namely sensory measurements of plant state with error. Thus our games are games with perfect information on the control law transmitted by Control to Plant, but with imperfect information on Plant state transmitted to Control by Plant.

Our game approach is different from traditional methods of extracting control in the presence of disturbances or measurement uncertainties. For example, one traditional control engineering approach is to start instead with a deterministic plant model which does not incorporate either disturbances or measurement uncertainties, to proceed to extract a suitable control program for the deterministic model, and afterwards to determine the effect of small changes in measurements and parameters on observability, controllability, and stability of the hybrid system. Another approach is to model the Plant by stochastic differential equations in the first place, and to look for stochastic control programs with optimal control features. A third approach is to use a two person differential game between Control and Plant or between Control and Disturbance. This usually entails extracting continuous control strategies which change control values continuously, based on continuous measurements of plant state. To extract such a continuously sensing continuously controlling strategy using differential games usually requires elaborate mathematical apparatus when it is possible at all. Our games approach differs from all three. Control strategies are not derived directly from a deterministic model. The model does not involve stochastic processes. It is a game approach, but not the usual differential games approach. In our games, one player, Plant, is constrained to follow a differential or difference equation guided by controls and subject to disturbances. The change is that, in our games, measurements of Plant state are communicated to Control only at discrete prescribed times, while a change in the control function imposed by Control on the Plant can be imposed instantaneously. The changes imposed by Control on the plant are event-driven based on the current state of the control automaton and the current measurement of plant state. Restricting Control in this way is natural if Control is to be a digital program, since a digital program changes its state based on a discrete sequence of successive input symbols representing plant measurements. Even if Control is not restricted to a digital program with finite alphabets and states, the discrete sensing, discrete mode switching control strategies turn out to be useful as intermediate idealized programs to extract before refining them to finite state strategies which give controllable-observable behavior.

The system model underlying our game is the hybrid systems model of Kohn-Nerode, [24] and [25], to which the reader is referred. The games approach stems from the Nerode-Yakhnis-Yakhnis [38] formulation of extracting concurrent programs as solving an appropriate game. The hybrid systems games were first announced in

Nerode-Yakhnis [36],[37].

Control automata which sense plant state at discrete times but exercise control over the plant continuously, with only occasional mode switching, operate in the following way. Their input alphabets, internal states, and output alphabets can be any finite or infinite nonempty set. They can be regarded as non-deterministic automata operating in continuous time. They change their input alphabet letter and internal state instantaneously at a discrete sequence of time instants only, being in the previous automaton state in a non-empty open interval preceding each such moment. These are the moments when sense data about the plant are communicated to the control automaton. Only at these times does the control automaton instantaneously change its output letter, called a control order. This output letter is to be interpreted in applications as a control order to the plant physical controller to change the control law used in that physical controller. For instance, in Kohn-Nerode extraction procedure [26, 15], this issued control order is a chattering control implemented via a finite sequence of "primitive" control actions, each specifying a physical controller parameter to be used for some period. Such a control order is a finite sequence of infinitesimal generators of flows. Each flow is to be followed in the prescribed order for a prescribed relative duration of the interval of time over which this control order persists.

In summary, control orders, or mode switches, are issued by the control automaton on an event driven basis based on past sense measurements of plant state. Although we allow the set of control automaton states to be infinite, in all our examples the automaton will be finite state, while the input alphabet representing possible sense measurements will be infinite.

Next we describe the underlying assumptions on the plant model and the control automaton for our basic discrete sensing game.

We assume as a physical realizability requirement that the discrete times at which the the control automaton issues control orders, $t_0 < t_1 < t_2 < t_3 < ....$, have a positive lower bound for the differences $t_{i+1} - t_i$. This usually called the Zeno requirement. We call these sequences admissible time sequences. In this section, we shall assume that for all i, $t_{i+1} - t_i = T$ is a fixed positive constant T. In a later section, this simplification is dropped.

Plant model

Our basic assumption of the plant model are the following.

1. We assume the plant is modeled by an ordinary vector differential equation

$$\dot{y}(t) = f(t, y(t), u(t), d(t)),$$

   where $y(t)$ is the plant state, $u(t)$ is a control function, and $d(t)$ is disturbance function.
2. The time t will range over the real interval $[0, \infty)$. Plant state trajectories $y(t)$, control functions of time $u(t)$, and disturbance functions of time $d(t)$ will be defined on $[0, \infty)$.
3. The function $y = y(t)$, which we call the plant state trajectory, takes values in X, the set of plant states. There will be a class $\mathcal{S}$ of admissible plant state trajectories.

4. The function $u = u(t)$ takes its values in a set U of admissible control values. There will a class $\mathcal{C}$ of admissible control functions.
5. The function $d = d(t)$ takes its values in a set D of admissible disturbance values. There will be a class $\mathcal{D}$ of admissible disturbance functions of time.
6. The sets of admissible plant states, control values, and disturbance values are assumed to be subsets of fixed finite dimensional Euclidean spaces.

Here is the kind of problem we want to solve. Suppose that a subset V of the plant states is specified, which we call the *viability set.* Suppose that a subset of the viability set V is given, which we call the *goal set* G. We want to extract a control strategy which satisfy the following conditions.

1. Starts the plant at time t_0 in a prescribed plant state y_0 in the viability set V.
2. Ensures that at all subsequent times t, the plant state $y = y(t)$ is also in the viability set V.
3. Ensures, as a winning condition for the game, that the plant state enters the goal set G by a designated time. (Alternative winning condition might that the plant state eventually enters the goal set G or the plant state must enter G in a certain time interval (t_{f_1}, t_{f_2}).)

All the control automaton can do at time t is to define the control law for the next interval to be incorporated into the control function of time. But the control automaton has no influence over the disturbance function of time $d = d(t)$ encountered. Thus the control automaton must select the next control law in such a way as to keep the plant state in the viability set V and lead to the goal, at a designated time or eventually, as required, no matter what admissible disturbance function is encountered.

All the information the control automaton has available to decide what new control to impose is its own automaton state plus the current sensor measurements of plant state.

In summary, the problem is to construct a control automaton which, given both its current state and measurement of plant state at the end of the current interval, changes to a new state and outputs new control law to be used for the next interval such that if the plant state starts at time t_0 in the viability set V, with a prescribed initial control, the plant state trajectories stay entirely within V and either enters the goal set G by a prescribed time or alternately eventually enters the goal set G.

Admissible Control Functions

Assume that the set of admissible control functions $\mathcal{C}$ is a set of functions which contains a set of functions C_0 from $[0, 1]$ to U. If $a < b$, and c is a control law from C_0, then the corresponding control law on $[a, b]$ is defined as the function $c((t-a)/(b-a))$. Our minimal assumption on the set of admissible control laws $\mathcal{C}$ is the following.

> Suppose that u maps $[0, \infty)$ into U and there exists a sequence of times $t_0 < t_1 < t_2 < t_3 < \ldots.$ such that for every n, there is a function c in C_0 for which c corresponds to u restricted to $[t_n, t_{n+1})$. Then u is in $\mathcal{C}$.

We also assume a similar relation between the set D_0 of admissible disturbances mapping $[0, 1]$ to V, and the set of admissible disturbance functions $\mathcal{D}$ of time mapping $[0, \infty)$ to D. We do not specify exactly the closure conditions on $\mathcal{C}$ or on $\mathcal{D}$. In

some contexts, $\mathcal{C}$ is the set of all continuous functions, $\mathcal{D}$ is the set of all measurable functions, etc.

Uniqueness of Plant State Trajectories

We assume that each admissible control function and disturbance function gives rise to a unique plant state trajectory. That is, suppose the classes $\mathcal{C}$, $\mathcal{D}$ and the plant function f are given. We shall assume that our plant model satisfies the following condition.

> Given an admissible control function u, an admissible disturbance function d, an admissible plant state y_0, a time t_0, there exists a unique admissible plant trajectory function $y = y(t)$ with domain $[t_0, \infty)$ such that $y(t_0) = y_0$ and for all $t \geq t_0$, y satisfies
>
> $$\dot{y}(t) = f(t, y(t), u(t), d(t)).$$

Bounded Measurement Error

We assume that if y is a plant state and m is its measurement, then there exists an $e > 0$ such that $|y - m| \leq e$.

We are now in position to define the legal positions of the a discrete sensing game. Assume that we are given a fixed admissible time sequence $t_i = t_0 + i\Delta t$.

Game Positions

Each (legal) position in the game will be a sequences of moves

$$m_0, c_0, m_1, c_1, ..., m_n, c_n$$

alternating between the players, with Plant moving first. Plant makes even numbered moves. Control makes odd numbered moves. Here is the simultaneous inductive definition of the notion of (legal) positions of the game, and of the trajectory associated with a position.

1. Suppose that p is the opening (null) position. Plant may choose as a move any admissible Plant state m_0. We call any admissible state x such that $|x - m_0| < e$ a trajectory associated with that position. That is, we interpret each such x as a possible measurement of true Plant state m_0 at time t_0, and also as a degenerate trajectory starting and ending at t_0.
2. Suppose that p is a position p of odd length. Control may choose as move any admissible control law c from C. The trajectories associated with position pc are the same as the trajectories associated with p.
3. Suppose that $p = p'c$ is a non-null position of even length with c its last move made at time t_i. Inductively, suppose we have already defined the set of all plant trajectories associated with p'. Then Plant may choose as move at position p any m such there exists a trajectory associated with p whose end state z has $|z - m| < e$. Inductively, we define the trajectories associated with the position p, m as those trajectories extending at least one trajectory associated with p to a trajectory defined also on $[t_i, t_{i+1}]$ which solves on that interval the same differential equation, using the control function of time on that interval associated with c and using some disturbance function of time on that interval associated with an admissible disturbance. Thus for any n, if Control makes

move c at time t_n, then the control function of time applied to the Plant in $[t_n, t_{n+1}]$ is $c(t-t_n)/(t_{n+1}-t_n))$. If d is in D_0, the corresponding disturbance function of time on the time interval $[t_n, t_{n+1}]$ is $d((t-t_n)/(t_{n+1}-t_n))$. Due to our trajectory field assumption, on $[t_n, t_{n+1}]$, there is a unique plant trajectory $y = y_{x,c,d,t_n}$ determined by the plant state $x(t_n)$ on the trajectory $x(t)$ associated with p, together with the control law c from C_0 and the disturbance d from D_0 and the differential equation.

$$\dot{y}(t) = f(t, y(t), u(t), d(t)).$$

For any x (in a Euclidean space) define $Ball(x, e) = \{y \in E^n : |x-y| \leq e\}$. For any subset Y the space, define $Ball(Y, e) = \cup_{x \in Y} Ball(x, e)$ The plant moves m can then be described as

$$\{z \in Ball(PlantStates, e) \mid (\exists\ v \in D)(|y_{x,c,v,t_i}(t_{i+1}) - z| \leq e\}.$$

We define the set of finite plays of the game to be the set of legal positions described above. An infinite play is an infinite sequence, each finite initial segment of which is a finite play. Trajectories associated with infinite plays are similarly defined.

There are alternate definitions of "winning the game", depending on what control problem has to be modeled. For example, given the basic control problem of trying to bring the plant from some initial point x in the viability set V to a point in the the goal set, G, the appropriate notion of "winning the game" is as follows.

Winning a Play

We say that Control wins play μ, or alternately that μ is a winning play for Control, if

1. μ is a finite play.
2. For the last Plant move m of μ, $Ball(m, e)$ is a subset of the goal set G.
3. All states traversed along all plant trajectories associated with μ are in the viability set V.

We note that there are other natural notions of winning plays depending on the control problem to be solved. For example, we might define Control as winning a play if all associated plant trajectories stay in an ϵ neighborhood of a fixed curve in plant state space. For example if $\phi(t)$ is optimal plant trajectory with respect to some Lagrangian L, then we might take the viability set V for this example as the set of pairs (x, t) such that x is a plant state and t is a time and $|x - \phi(t)| \leq \epsilon$. Our games can easily be modified to deal with a variety of control problems.

A *strategy* for Control is a map F from the set of positions of the game of odd length into C_0. The idea here is that, given a play $m_0, c_0, m_1, c_1, ..., m_n$, the function $F(m_0, c_0, m_1, c_1, ..., m_n) = c_n$ determines the next move of Control. We say that a play $P = m_0, c_0, m_1, c_1, ..., m_n, c_n$ is generated by the strategy F, or that p is play in which Control follows F, if for all i, $c_i = F(m_0, c_0, m_1, c_1, ..., m_i)$. Strategies for Plant can be defined in a similar manner.

The notion of which strategies are winning for Control depends on the definition of what it means for Control to win the game. In the remainder of this section a strategy F for Control is a *winning strategy* if, whenever Control follows F, Control

will **eventually** reach a winning position, no matter what initial position m_0 in the viability set V is chosen by the Plant to start the game and no matter what the subsequent moves of Plant are.

An *automaton strategy* for Control is an automaton with the following properties.

1. The set of automaton states S is any non-empty set.
2. The automaton input alphabet is $Ball(V, e)$ where V is the viability set.
3. The automaton output alphabet is C_0.
4. The automaton transition table $M(s, m)$ and its output function $H(s, m)$ are such that the output is produced simultaneously with the automaton shifting to its new state $r \in M(s, m)$.

We call such an automaton a control automaton.

We say that a Control automaton strategy generates a play

$$\mu = m_0, c_0, m_1, c_1, ..., m_n ..., c_n.$$

if

1. $c_0 = H(s_0, m_0)$ and the next control automaton state is $s_1 = M(s_0, m_0)$ where s_0 is the initial state of the automaton.
2. At time $t_k = t_0 + kT$ in a position with last Control move c_k and last Plant move m_k, the next control automaton state is $s_{k+1} = M(s_k, m_k)$ and the next control law is $c_{k+1} = H(s_k, m_k)$.

We say that an automaton strategy for Control, or equivalently control automaton, is *winning* for Control if whenever Control generates plays following the control automaton as described above, then Control will reach a winning position, no matter what initial move m_0 in the viability set V is chosen by the Plant to start the game, and no matter what the subsequent moves of Plant are.

Finite Input-Output Alphabet Games

Real digital controllers are finite state machines with finite input and output alphabets. We adapt our definitions for using such controllers as Control strategies. First let V' be a finite subset of $Ball(V, e)$ such that

$$(\forall\, y \in V)(\exists y' \in V')\ (|y - y'| \leq \delta).$$

Then if we we replace V by V' in all the definitions above and we assume that the set of controls C_0 is finite, then we have defined a subclass of games which we call finite alphabet discrete sampling games. For these games the control automata are always finite automata.

We end this section by giving an explicit example of how a problem that has been studied in the literature can be expressed in game language.

Railroad Problem: This is a variation of a problem considered by Schneider and Marzullo [32]. Here the plant is a train whose plant state space consists of pairs (y, ζ) where y is a position on a line and ζ is the train velocity at that position. Thus

the plant space is a subset of a 2- dimensional Euclidean space. The plant dynamics are given by

$$\begin{cases} \dot{y} = \zeta \\ \dot{\zeta} = u + v \end{cases}$$

where u is a control parameter and v is the train engine acceleration. Sensors can measure the train position and velocity with known error bounds. We assume that there is a common bound e on uncertainty in position and velocity. There is a viability set V based on a partition of the track into contiguous blocks. For each block, there are regulations requiring that certain minimum and maximum velocity bounds be respected when the train is on that block. That is, suppose there are $n \geq 1$ blocks, and each block is defined by its beginning position and its length (b_i, ln_i), $0 \leq i \leq n-1$. The corresponding velocity bounds are (min_i, max_i). Thus

$$V = \{(y, \zeta) \mid b_i \leq y \leq b_i + ln_i \Rightarrow \zeta \in [min_i, max_i], 0 \leq i \leq n-1\}.$$

The velocity is assumed to be in a fixed direction along a straight railroad line. Hence all positions of the train are in that direction from the initial position 0.

The goal set is defined by a distance $D > 0$ from the origin where

$$D \leq \sum_{0 \leq i \leq n-1} ln_i.$$

That is,

$$G = Ball((D, e) \times \{0\}.$$

The problem is to guide the train to stop within the interval $[D - e, D + e]$ while satisfying the blocks constraints along the way.

3 Continuous Sensing, Discrete Mode Switching

In this section, we define a second class of games which we call continuous sensing games. Throughout this section, we keep the same set of assumptions on the plant model and continue that same notation as used in section 2. Our basic underlying model is a hybrid system in which the plant state is sensed continuously, but new control orders (mode switches) are issued at discrete times. We illustrate this idea with the following simple example.

Water Level Monitor

Our water level monitor is a generalization of an example analyzed in [1]. The plant consists of a water pump and a water tank. The controller issues control orders to turn on or turn off to the pump. The plant state is the pair consisting of the water level $y \geq 0$ and the state of the pump $pmp \in \{on, off\}$, telling whether the pump is on or off. The state of the pump determines the dynamics of the water level. We assume that the water level y satisfies

$$\dot{y} = \begin{cases} f_1(y) \text{ if the pump is on} \\ f_2(y) \text{ if the pump is off} \end{cases} \quad (1)$$

where f_1 and f_2 are continuous functions such that

$0 < a' < f_1(y) \leq a$, for all y and
$0 > -b' > f_2(y) \geq -b$, for all y.

Moreover, we shall assume that there are constants L_1 and L_2 such that for all x and y, $|f_i(x) - f_i(y)| \leq L_i|x - y|$ for $i = 1, 2$.

Thus the states of the plant can naturally be partitioned into two disjoint classes; one class where the pump is on and the other class where the pump is off. The controller has two control actions $\{pon, poff\}$ which cause transitions between the two classes of plant states. We assume that the transitions take time up to $d > 0$, the delay, to complete. That is, until a transition has been completed, the pump is regarded as being in its preceding state and the corresponding equation for the water level dynamics applies.

Our controller has only two states: $\{son, soff\}$. The action of the controller is the following. If the controller receives a measurement y of the current water level when the controller is in state *son*, then it checks whether the condition $y \geq g$ holds where $g > 0$ is a given constant. If the condition holds, then the controller outputs a order $poff$ to cause the pump to be turned off and the controller shifts to the state $soff$ instantaneously. Otherwise, the controller remains in its state *son* and outputs no order to the pump. If the controller is in the state $soff$ and receives a measurement y, then it checks whether the condition $y \leq h$ holds where $h > 0$ is another constant. If the condition $y \leq h$ holds, then the controller outputs the order *pon* to cause the pump to be turned on and instantaneously shifts to the state *son*. Otherwise, the controller remains in the state $soff$ and outputs no order to the pump.

We note that while the controller instantaneously shifts to a new state, the pump does not instantaneously change its corresponding state, so the controller may lose the natural correspondence between its state and the state of the pump. Note also that the controller is not digital, since it is expected to act at the exact instant when the water level satisfies the conditions causing the controller to shift states and the water level is measured continuously.

The controller and the plant interact forever. We wish to find those values of (g, h) which will guarantee that the water level is maintained forever between two constants $0 < u < v$. That is, we want to design our controller to pick (g, h), so that at all times t, $u \leq y(t) \leq v$.

Formally, a plant state is a pair $(y(t), z(t))$, where $y(t)$ is the water level, $z(t) = 1$ if the pump is on at time t, and $z(t) = 0$ if the pump is off at time t. The control parameter takes on only two values, 0 and 1, where 0 indicates that the pump has been told to turn off and 1 indicates that the pump has been told to turn off. There is no disturbance. The space of control laws is the set of piecewise constant functions with values in $U = \{0, 1\}$. The dynamics of the plant has a form given by (1). The conditions on the f_i which ensure that the systems always has unique fully extendible trajectories for any given initial condition, given at the end of the section, are satisfied.

This ends temporarily our discussion of the the water tank example. We go on to the definition of a general class of games which will describe examples like this one.

Next we present two equivalent game models for continuous sensing games.

Game Model I

We begin with a plant given by an ordinary differential equation with control and disturbance. We consider the set of plant trajectories that begin at a time t_0 at a points x_0 and satisfy the plant dynamics described at the previous section for some admissible set of control functions $\mathcal{C}$ and some admissible set of disturbance functions $\mathcal{D}$. We write $TRAJ$ for the set of all functions $Y(t) : [t_0, \infty) \to X$ such that there exists a control function $u \in \mathcal{C}$ and a disturbance function $v \in \mathcal{D}$ such that

1. $Y(t_0) = x_0 \in X$ and
2. $\dot{Y}(t) = f(t, Y(t), u(t), d(t))$ for all $t \geq t_0$.

We are assuming that there is a unique $Y \in TRAJ$ corresponding to any choice of t_0, $Y(t_0)$, $u(t)$, and $d(t)$.

Here is the game. There are two players: Plant and Control. Plant moves are taken from X but their choice is governed by certain members of $TRAJ$. Control moves are taken from the set $(\mathcal{C} \cup \{no\ action\}) \times [0, \infty)$. Suppose the game starts at time t^*. The exchange of moves between Plant and Control results in a function of time

$$\mu(t) = \langle y(t), z(t) \rangle$$

where $y(t) \in X$ and $z(t) \in (\mathcal{C} \cup \{no\ action\}) \times \{t\}$. A value of such function at time t is a pair of the last plant state $y(t)$ observed and the corresponding Control move which we regard here as occurring instantaneously. To determine its next move $z(t)$, Control may utilize all values of $y(\tau)$ at all times τ up to and including t. We call such a function μ a play, if the following is true. There is a strictly increasing sequence of times $\{t_k : k \geq 0, t_0 = t^*\}$ such that for every $k \geq 0$, $z(t_k) \in \mathcal{C} \times \{t_k\}$ and for every $t \in (t_k, t_{k+1})$, $z(t) = (no\ action, t)$ and $y(t) = Y_k(t)$, where $Y_k(.)$ is a member of $TRAJ$ determined by $t_k, x_0 = y(t_k)$, the control law in $z(t_k)$, and some admissible disturbance $d(t) \in D$. That is, if $z(t_k) = (u_k(.), t_k)$, then $Y_k(t)$ satisfies that $Y_k(t_k) = y(t_k)$ and $\dot{Y}_k(t) = f(t, Y_k(t), u_k(t), (t))$ for all $t_k \leq t$ for some admissible disturbance function $d(t)$. We call the moves of Control at the times t_k for $k \geq 0$ *essential Control moves* and the moves at the time $t \notin \{t_k : k \geq 0\}$, i.e. where $z(t) = (no\ action, t)$, *inessential Control moves.* If the sequence $\{t_k : k \geq 0, t_0 = t^*\}$ is finite with the last index being n, we put $t_{n+1} = \infty$ and the above definition of a play applies.

Definition 1. Call a sequence $\{t_k : k \geq 0\}$ **realizable** if

$$inf\{(t_0 - t^*), (t_{k+1} - t_k) : k \geq 0\} > 0.$$

Call a play of the game, **realizable** if the sequence of instances of essential Control moves in it is realizable. We will consider that the plays which are not realizable are lost by Control.

Game model II

Next we will describe the plays in our continuous sensing game in a slightly different but equivalent way in order to bring out the resemblance with those games in which moves alternate in discrete time. Only the essential control moves will be

displayed in plays. Assume that, at the start of the game the time is $t^\star$, the plant state is $x^\star \in X$, and the initial control law is $u^\star(t)$. We define a block to be a contiguous segment of a play over a right-open interval of time where the corresponding Control moves are inessential except for the leftmost Control move. A block may be infinite if there is no essential Control move after it. In presenting a block, we suppress the inessential Control moves in it and we specify the moves of Plant by giving the element $Y \in TRAJ$ that determines its moves in the segment. We remove the Control move from the leftmost pair of moves in the block and place it in front of the block not regarding it as a part of the block. A play of the game is thus represented as a sequence of blocks alternating with a sequence of essential Control moves. Finite sequences of this sort are called the positions of the game. A play is a sequence of positions such that each next position extends the preceding one. We will describe all admissible positions by means of induction on the length of positions. We will simultaneously define by induction a segment of the plant trajectory corresponding to a position in the game. Thus we will define, by induction on n, the positions p_n, the plant trajectory segment γ_n corresponding to the position p_n, and the right ends $t(n)$ of the domains of γ_n.

(1) $n = 1$.

Then we let $p_1 = \langle u^\star(.), t^\star \rangle$. The corresponding segment γ_1 of the plant trajectory is a single point $(t^\star, x^\star)$, i.e. $\gamma_1(t^\star) = x^\star$. We denote the right end of the time interval of γ_1 as $t(1) = t^\star$.

Next suppose that the positions p_{n-1} corresponding to $n-1$ successive admissible moves are defined along with the corresponding plant trajectory γ_{n-1} which is defined over the interval $[t^\star, t(n-1)]$.

(2) $n = 2 \cdot k + 2$.

Suppose $p_{n-1} = \langle u^\star(.), t^\star \rangle \cdot B_0 \cdot z_0 \cdot \ldots \cdot B_{k-1} \cdot z_{k-1}$. Then the next admissible block B_k of Plant moves is specified by any member $Y_k : [t(n-1), \infty) \to X$ of $TRAJ$ such that $Y_k(t(n-1)) = \gamma_{n-1}(t(n-1))$ and satisfies

$$\dot{Y_k}(t) = f(t, Y_k(t), u_{k-1}(t), d(t)) \text{ for all } t \geq t(n-1)$$

where $u_{k-1}(.)$ is the control law $u_{k-1}(.)$ that occurs in the last Control move z_{k-1} and $d(.) \in \mathcal{D}$. The plant trajectory corresponding to the p_n is the function $\gamma_n : [t^\star, \infty) \to X$ defined by

$$\gamma_n(t) = \begin{cases} \gamma_{n-1}(t) \text{ if } t \in [t^\star, t(n-1)] \\ Y_k(t) \quad \text{if } t > t(n-1). \end{cases}$$

(3) $n = 2 \cdot k + 3$.

Suppose $p_{n-1} = \langle u^\star(.), t^\star \rangle \cdot B_0 \cdot z_0 \cdot \ldots \cdot B_{k-1} \cdot z_{k-1} \cdot B_k$. Then a position of length n extending p_{n-1} is of the form, $p_n = p_{n-1} \cdot z_k$ where z_k is any Control move of form $\langle u_k(.), t_k \rangle$ such that $t_k > t_{k-1}$ and $z_{k-1} = \langle u_{k-1}, t_{k-1} \rangle$. We then put $t(n) = t_k$ and γ_n equal to γ_{n-1} restricted to the interval $[t^\star, t(n)]$.

An infinite sequence of positions, linearly ordered by extension, defines an infinite play. All finite sequences of the form

$$\langle u^\star(.), t^\star \rangle \cdot B_0 \cdot z_0 \cdot \ldots \cdot B_{k-1} \cdot z_{k-1} \cdot B_k$$

are plays too. Plays which are not realizable in the sense of definition 5.1 are regarded as lost by Control.

It is easy to see that there is a natural bijection between the plays of Game Model I and the plays of Game Model II.

Remark For convenience of notation, we will suppress the symbols for blocks B_k of Plant moves and use instead the plant trajectory Y_k which specifies the block. We also suppress the first move in the plays described inductively above because we regard it as fixing the game. That is, the initial move simply corresponds to giving initial settings of the plant, including initial control parameter values. Thus we will denote a play by a sequence of the form

$$Y_0, z_0, Y_1, z_1, ..., Y_k, z_k, ...$$

According to our definition, each of the trajectories $Y_i : [t_{i-1}, \infty) \to X$ is infinite. (Here we make the convention that $t_{-1} = t^*$). Of course, in the case when there is another essential control move after time t_{i-1}, we only use the finite trajectory segment,$y_i = Y_i$ restricted to $[t_{i-1}, t_i]$, to determine the final plant trajectory. Thus an even more compact notation for a play is a sequence of the form

$$y_0, z_0, y_1, z_1, ..., y_k, z_k, \ldots.$$

We note, however, that this last notation could be misleading since it makes it appear that the time of the next essential move of Control is part of the previous move of Plant. A move of Plant does not force the timing of the next essential Control, this is forced by Control's strategy.

A winning condition for Control is a set of realizable plays whose corresponding plant trajectories γ satisfy the performance specification imposed on the hybrid system. For example, in the water level game, the performance specification is that for all t, $u \leq y(t) \leq v$.

We are interested in existence of winning strategies for Control in such a game. Intuitively, a strategy is any kind of systematic behavior of Control in a game which determines its next move on the basis of the knowledge of the previous moves of the players in a play. A winning strategy is a behavior that is

1. defined for all positions which are reached while using such a behavior and
2. all plays generated by such a behavior are winning for Control.

Following Buchi, we consider a description of such a behavior by means of an automaton whose input alphabet is the set of the opponent's moves X, and whose output alphabet is $(\mathcal{C} \cup \{no\ action\}) \times [0, \infty)$. We do not require at this point that either of the alphabets be finite and we do not require that the automaton set of states be finite.

Such an automaton is to be capable of continuously reading its input. At the end of this section, we give a formal definition of a continuous input-discrete output automaton and describe sufficient conditions for such an automaton to generate exclusively realizable plays.

Modeling Delays We can model a delay in resetting the next control law to be imposed on the plant. Such a delay may depend on the current control and on the next control law. We assume that the two laws determine an upper bound d

for the reset time interval. We model this in the game rules for Plant. If Control makes a move $\langle u_k(.), t_k \rangle$, we view this as an order to reset the current control law to $u_k(.)$. The actually time τ_k at which we change to the new control law will be some $\tau_k \in [t_k, t_k + d]$. The Plant moves which form the next block will be of the form

$$Y_k(t) = \begin{cases} Y_{k-1}(t) & \text{for } t_k \leq t \leq \tau_k \\ Z_k(t) & \text{for } t \geq \tau_k \end{cases}$$

where Z_k mapping $[\tau_k, \infty)$ into the plant states is the unique trajectory determined by the initial condition $Z_k(\tau_k) = Y_{k-1}(\tau_k)$, the control law $u_k(.)$, and an admissible disturbance function $d(.)$. All the rest is as in the preceding definition of the game, except that it is the reset times, rather than the time Control moves, which determine the plant trajectory corresponding to a play. The realizability of a play is determined by the sequence of reset times, so we must assume that for any k, $t_{k+1} - t_k$ exceeds the positive lower bound d.

3.1 Uniqueness and Extendibility of Plant Trajectories

Next we discuss sufficient conditions for the plant trajectory corresponding to a play to be unique and continuous. For an example of such a condition, consider the Caratheodory conditions ([12]) to be imposed on the plant model f modified to allow control and disturbance parameters.

Caratheodory Conditions

We consider plants modeled by the vector ordinary differential equation

$$\dot{y} = f(t, y, u, d)$$

where $t \in [t_0, \infty)$, $x \in X$, $u \in U$, $d \in D$, and which satisfy the following conditions.

CC 1: For every u and for almost all t, the function $f(t, x, u, d)$ is continuous in (x, d).
CC 2: For every u and every x, the function f is measurable in (t, d).
CC 3: For every u, there is function $m(.)$ over $[t_0, \infty)$, which is Lebesgue integrable over every finite interval of its domain and such that $|f(t, x, u, d)| \leq m(t)$ in $[t_0, \infty)$ for every x and d.

Theorem 2. *Suppose that the admissible control laws are piecewise constant over time, that disturbances are measurable functions over time, and that the plant dynamics f satisfy Caratheodory conditions* **CC 1-CC 3**. *Suppose also that*

1. *The plant state space X coincides with the Euclidian space containing it.*
2. *For every $u \in U$, there is a function $L(.)$ over $[t_0, \infty)$, which is Lebesgue integrable over every finite interval of its domain, and such that $|f(t, x, u, d) - f(t, y, u, d)| \leq L(t) \cdot |x - y|$ for every x, y and d, and*
3. *For every u, there is a constant a such that $|f(t, x, u, d)| \leq a \cdot (1 + |x|)$ for all t, x and d.*

Then to every realizable play in the continuous sensing game described above, there corresponds a unique absolutely continuous plant trajectory defined over $[t^\star, \infty)$, where $t^\star$ is the time the play begins.

Proof. We show by induction on k the uniqueness and absolute continuity of the plant trajectory γ corresponding to a segment of a play up to time t_k and defined over the interval $[t^*, t_k]$. It is sufficient to do the inductive step. Assume that the statement is true for k.

(A) If there is no essential control move after t_k, consider

$$\dot{y}(t) = f(t, y(t), u_k, d(t))$$

for $t \geq t_k$. Here $d(.)$ is a measurable disturbance that occurs in the plant for $t \geq t_k$, u_k is the value of the constant control function which is part of the essential Control move at t_k. We have to show existence, uniqueness of an absolute continuous function satisfying the differential equation for $t \geq t_k$ and beginning from the point $\gamma(t_k)$. This would yield an absolutely continuous extension of the plant trajectory realizing the trajectory corresponding to a play. By assumption (1), we may assume that $\gamma(t_k) \in X$. We have to check only that $F(t, y) = f(t, y, u_k, d(t))$ satisfies the standard Caratheodory conditions, a uniqueness of a solution condition, and an extendibility condition. The standard Caratheodory conditions CC 1-CC 3 are obtained from **CC 1- CC 3** by omitting control and disturbance parameters. We will check them for F.

Original Caratheodory condition CC 1: According to **CC 1**, there is a set E of the measure 0 of times such that for any t not in E, $f(t, y, u_k, d)$ is continuous in (y, d). Fix such a t. Then $f(t, y, u_k, d(t))$ is continuous in y. That is, $F(t, y)$ is continuous in y for all t not in E. This verifies the original condition CC 1.

Original CC 2: We need only show that for every y, $F(t, y)$ is measurable in t. By **CC 2**, $f(t, y, u_k, d)$ is measurable in (t, d) for every y. Since $d(.)$ is measurable and the composition of measurable functions is measurable, it follows that $f(t, y, u_k, d(t))$ is measurable in t for every y. This is the desired conclusion for F.

Original CC 3: The adapted **CC 3** gives the function $m(.)$ for f depending on u. So we take the m corresponding to u_k and it provides the desired bound for F.

From the standard Caratheodory conditions for F, it follows that there exist solutions of the equation $\dot{y}(t) = F(t, y(t))$ for every $t^{**} \geq t^*$, $x^{**} \in X$ in some interval $[t^{**}, t^{**} + p]$ where $p > 0$ and $y(t^{**}) = x^{**}$, see [12], page 4.

The uniqueness of F easily follows from assumption (2). That is, choose the function $L(.)$ according to (2) which corresponds to u_k. Then

$$|F(t, x) - F(t, y)| = |f(t, x, u_k, d(t)) - f(t, y, u_k, d(t))| \leq L(t) \cdot |x - y|.$$

It then follows that there is a *unique* trajectory of F passing through every point (t^{**}, x^{**}), see [12], page 5.

Finally we consider the extension of solutions of $\dot{y} = F(t, y)$. Since the standard Caratheodory conditions are satisfied by F, according to [12], page 7, every solution can be extended on both sides of an initial condition to the boundary of any closed and bounded domain of F. By condition (3), choose a to correspond u_k. Then $|F(t, y)| \leq a \cdot (1 + |y|)$. From estimates based on this condition, it follows that the states of any trajectory over a finite closed interval of time lie in a finite ball B whose radius depends only on size of the interval. Using (1), we may choose for any $t > t^*$, the domain for F to be $[t^*, t] \times B$. This is a closed and bounded domain. From the quoted theorem, it follows that a solution can be extended on the whole of $[t^*, t]$.

This completes the demonstration of existence of the unique absolutely continuous plant trajectory satisfying $\dot{y}(t) = f(t, y(t), u_k, d(t))$ for all $t \geq t_k$.

(B) There is an essential Control move at $t_{k+1} > t_k$. For a construction of a plant trajectory up to t_{k+1} corresponding to a play, we have to show that there is the unique absolutely continuous plant trajectory satisfying $\dot{y}(t) = f(t, y(t), u_k, d(t))$ for all $t \in [t_k, t_{k+1}]$. In this case, the proof is similar to the proof used for case (A). □

We note that the conclusions of the theorem hold, in particular, for f independent of time and if for all control values u, f is continuous in (x, d) and satisfies conditions (1)-(3) of the theorem.

3.2 Continuous Input-Discrete Output Automata.

Next we want to consider the analogues of an automaton winning strategy for continuous sensing games. For this purpose, we introduce continuous input-discrete output automata to represent strategies for Control in continuous sensing games.

We adopt the following definition of the behavior of an ordinary automaton in continuous time. First we define the notion of an automaton run on an input word as a function of continuous time. Recall the ordinary definition of a run for a finite state automaton. Let $x = x_0 x_1 \ldots x_n$ be an input word. Then a run $r = r(0), r(1), \ldots$ is the sequence of the automaton states satisfying

$r(0) = s_{in}$ and
$r(k+1) = M(r(k), x_k)$ for all $k \geq 0$

where s_{in} is the automaton initial state and M is its transition table.

Definition 3. Suppose $0 \leq t_0 < t_1 < \ldots < t_n$ is an increasing sequence of times at which the letters of an input word $x = x_0 x_1 \ldots x_n$ are read. A function $r : [0, \infty) \to S$ is a **run of the automaton in continuous time** if

$$r(t) = \begin{cases} s_{in} & \text{if } t \in [0, t_0] \\ M(r(t_k), x_k) & \text{if } n > k \geq 0 \wedge t \in (t_k, t_{k+1}] \\ M(r(t_n), x_n) & \text{if } t \in (t_n, \infty) \end{cases}$$

Definition 3 says that system trajectories, viewed in the automaton state space, are functions of time that are piecewise constant, continuous from the left at all times.

Continuously Reading Automata

We now introduce a definition of the input-output automata used to model controllers capable of continuously reading input that may continuously change. We adopt the view that transitions are instantaneous and that state transitions are continuous from the left for automaton runs which correspond to a continuous stream of input in time. We also restrict attention to the output of the automaton at a discrete sequence of times.

Definition 4. A continuous-input discrete-output automaton consists of
1. A nonempty set of states S,
2. A nonempty input alphabet I,

3. An output alphabet $J \cup \{no\ action\}$ where $J \cap \{no\ action\} = \emptyset$,
4. A transition table $M : S \times I \to S$,
5. An output function $H : S \times I \to J \cup \{no\ action\}$,
6. An initial state s_{in}.

For example, the controller described previously for water level translates into the following continuous input-discrete output automaton. Its set of states is $S = \{son, soff\}$, the initial state is $s_{in} = son$, the input alphabet is $I = \{y : y \geq 0, y \in R\}$, and the alphabet of essential outputs is $J = \{pon, poff\}$. Here R is the set of reals. The transition table and the output function are defined as follows:

$$M(son, y) = \begin{cases} soff & \text{if } y \geq g \\ son & \text{if } y < g \end{cases}$$

$$M(soff, y) = \begin{cases} son & \text{if } y \leq h \\ soff & \text{if } y > h \end{cases}$$

$$H(son, y) = \begin{cases} soff & \text{if } y \geq g \\ no\ action & \text{if } y < g \end{cases}$$

$$H(soff, y) = \begin{cases} son & \text{if } y \leq h \\ no\ action & \text{if } y > h \end{cases}$$

Definition 5. Let $TIME = [t^*, \infty)$, $r(0) = s_{in}$. Suppose that $x(.)$ maps $TIME$ into I. A run of a continuous-input discrete-output automaton corresponding to the input stream $x(.)$ is a function $r : TIME \to S$ such that for every $t \in TIME$, there is a duration $\tau > 0$ such that $r(t') = M(s, x(t))$ for every $t' \in (t, t + \tau]$. The output function of a continuous-input discrete-output automaton corresponding to the input stream $x(.)$ and a run r is the function $h : TIME \to J \cup \{no\ action\}$ defined by $h(t) = H(r(t), x(t))$.

Given a run r of a continuous-input discrete-output automaton A corresponding to an input stream $x(.)$, we define the set of *switching times* of r, $SW(r)$, to be the set of all $t \in TIME$, such that $M(r(t), x(t)) \neq r(t)$.

The difference between our continuous input-discrete output automaton and a standard Mealy machine is that we allow the input to be an arbitrary function of continuous time rather than a piecewise constant function of time which reflects input at discrete instants only. We call the subset J of the automaton output alphabet, the alphabet of *essential outputs.*

The definition of run for a continuous-input discrete-output automaton gives rise to piecewise constant and continuous from the left state space functions which represent the transitions. One of the reasons we adopt this definition is to avoid the difficulties associated with the following automaton. Let $S = \{0, 1\}$, $s_{in} = 0$, $I = [5, 10]$, $t_0 = 1$, and the automaton transition function be given by

$M(0, 5) = 1$
$M(1, x) = 0$ for $x > 5$.

Suppose the input function is $x(t) = 5 \cdot t$. Then a transition should occur at $t = t_0$. However if such a transition does occur, then at *any* later instant $t_1 > t_0$ where the new state is $s = 1$ another transition from the state $s = 1$ back to the state $s = 0$ must occur. So at some $t_2 > t_1$, the automaton is again in state $s = 0$. This would imply that such transition times occur arbitrarily close to t_0. But this is inconsistent with our intuition of an automaton transition while continuously reading the input because there is no finite interval of the form $(t_0, t_0 + \tau]$ during which the automaton is in a fixed state.

Definition 6. Assume that the automaton input alphabet I is a subset of a Euclidean space. Call the sets $G_s = \{i : M(s,i) \neq s\}$ the switching sets. We say that a continuous input-discrete output automaton A has **separated switching sets** if for every s, and s', the Euclidean distance between the sets G_s and $G_{s'}$ is positive, i.e. $\rho(G_s, G_{s'}) > 0$. Here $\rho(G_s, G_{s'}) = inf\{\rho(x,x') : x \in G_s,\ x' \in G_{s'}\}$ where $\rho(x,x')$ is the usual Euclidean distance function.

Theorem 7. *Consider a continuous input automaton with the following properties.*

(a) Its set of states is finite.
(b) Its input alphabet is a subset of a Euclidean space E.
(c) Its switching sets of inputs are separated.
(d) Its switching sets of inputs are closed in the subset topology of E.

Then for every input function $x(.)$ which is continuous over $TIME = [t^*, \infty)$, there is a unique run of the automaton over $x(.)$. Moreover, the set $SW(r)$ of switching times during the run is discrete with no limit points in $TIME$.

Proof. Clearly, if $G_{s_{in}}$ does not intersect the range of $x(.)$, then $r(t) = s_{in}$ for all $t \in TIME$. In this case, SW is empty and clearly the conclusions of the theorem are satisfied. Suppose $G_{s_{in}}$ does intersect the range of $x(.)$. Then $x^{-1}(G_{s_{in}}) \neq \emptyset$. This set is also closed, since by assumption $x(.)$ is continuous and $G_{s_{in}}$ is closed. Hence, there is the least time t_0 such that $t_0 \in x^{-1}(G_{s_{in}})$. This is the first switching time. We include t_0 in SW. We associate the state $s_0 = s_{in}$ with t_0.

Next suppose we have constructed an increasing sequence of switching times $t_0, \ldots, t_k$ and the sequence of the corresponding states up to $s_0, \ldots, s_k$ at these switching times. Consider $s_{k+1} = M(s_k, x(t_k))$. Then either $x^{-1}(G_{s_{k+1}}) \cap (t_k, \infty) = \emptyset$, in which case $r(t) = s_{k+1}$ for all $t > t_k$, or $x^{-1}(G_{s_{k+1}}) \cap (t_k, \infty) \neq \emptyset$. If $x^{-1}(G_{s_{k+1}}) \cap (t_k, \infty) \neq \emptyset$, then the set $x^{-1}(G_{s_{k+1}}) \cap [t_k, \infty)$ is closed. Moreover t_k cannot be a limit point of $x^{-1}(G_{s_{k+1}}) \cap [t_k, \infty)$. That is, if t_k were such a limit point, there would be a sequence of points t'_j in $x^{-1}(G_{s_{k+1}})$ converging to t_k. But then, because of the continuity of $x(.)$, it must be that $x(t_k)$ is a limit point of a sequence $x(t'_j)$ of points from $G_{s_{k+1}}$. This would contradict the separateness of $G_{s_{k+1}}$ from the switching set G_{s_k} containing $x(t_k)$. Then we let t_{k+1} be the least element of $x^{-1}(G_{s_{k+1}}) \cap [t_k, \infty)$.

Thus by induction, we can define two sequences $\{t_k\}$ and $\{s_k\}$ such that for all k,

$$t_{k+1} \in x^{-1}(G_{s_{k+1}}) \cap (t_k, \infty)$$

and

$$s_{k+1} = M(s_k, x(t_k)).$$

Let SW be the set of elements in the first sequence. We claim that SW has no finite limit points in $TIME$. Indeed suppose the sequence t_k converges to $t^{**} > 0$. Since the set of states S is finite, there is a strictly positive number $\alpha = min\{\rho(G_s, G_{s'}) : s \neq s', s, s' \in S\}$. Choose $\varepsilon < \alpha/2$. By continuity of x, there exists an $\delta > 0$ such that $|t - t^{**}| < \delta$ implies $|x(t) - x(t^{**})| < \varepsilon$. Consider k_0 such that for every $k > k_0$, $|t_{k+1} - t_k| < \delta$. Then for all such k

$$|x(t_{k+1}) - x(t_k)| < 2 \cdot \varepsilon = \alpha$$

However for all k, $x(t_k) \in G_{s_k}$ and by the separateness of the switching sets, it follows that

$$|x(t_{k+1}) - x(t_k)| > \alpha.$$

This is a contradiction and hence the set $SW(r)$ has no finite limit points.

Since no transitions are possible at times between switching times, we have, besides the constant run mentioned above, two more types of runs depending on whether the set of switching times SW is finite or infinite. If SW is finite with last switching time t_n, then set

$$r(t) = \begin{cases} s_{in} & \text{if } t = t_0 \\ M(r(t_k), x(t_k)) & \text{if } n > k > 0 \wedge t \in (t_k, t_{k+1}] \\ M(r(t_n), x(t_n)) & \text{if } t \in (t_k, \infty). \end{cases}$$

If SW is infinite, then set

$$r(t) = \begin{cases} s_{in} & \text{if } t = t_0 \\ M(r(t_k), x(t_k)) & \text{if } k \geq 0 \wedge t \in (t_k, t_{k+1}] \end{cases}$$

The uniqueness of runs follows by induction on the switching times.

□

Next, we single out a property of a continuous-input discrete-output automaton which has been proved in the previous proposition, but can be established with a slightly weaker assumption. We will use this fact later in this section.

Proposition 8. *Suppose that the premises of Theorem 7 hold, but the requirement that the set of the automaton states be finite is omitted. Then at any state at which the automaton is continuously reading a continuous function $x : TIME \to I$, either the automaton remains in this state forever or there is a finite time $t > 0$ at which a transition to a different automaton state takes place.*

Definition 9. A run r of a continuous-input discrete-output automaton is **realizable** if

(a) Both transitions to new states and essential outputs, occur only at discrete times $DT = \{t_0 < t_1, < \ldots\}$. That is, $M(r(t), x(t)) = r(t)$ and $H(r(t), x(t)) =$ *no action* for t not in DT and $r(t_k) \neq r(t_{k+1})$ and $H(r(t_k), x(t_k)) \in J$ for $k \geq 0$.

(b) $r(t) = M(r(t_k), x(t_k))$ for every $k \geq 0$, $t \in (t_k, t_{k+1}]$.
(c) $inf(\{(t_{k+1} - t_k) : k \geq 0\}) > 0$.

If the sequence DT is finite and n is the last index k occurring in it, define $t_{n+1} = \infty$. Then (a)-(c) of the definition apply to the interval (t_n, t_{n+1}).

For example, the continuous input-discrete output automaton representing the controller for the water pump given above satisfies the premises of the Proposition 7. Therefore it has runs over continuous water level trajectories $y(.)$. Moreover, the proposition tells us that these runs are realizable.

Conditions (a) and (b) given in the definition of a realizable run reflect our intuition of automaton transitions as described above. The definition synchronizes transitions to new states with essential outputs. For automata with separated switching sets which satisfy the other conditions of Theorem 7, once this synchronization is present, every run is automatically realizable.

Condition (c) prevents the set of transition times DT from having finite limit points. We call (c) a realizability condition. The definition of realizable run here has the same motivation as that of realizable time sequence in [24] and of "bounded from below" sampling intervals for a controller in [36].

By taking $I = X \times [0, \infty)$, we can make the automaton transition table and the output function depend explicitly on time, $M(s, x, t)$; $H(s, x, t)$. We can then ensure that the conditions (a)-(c) are satisfied for all runs of the automaton over any input function $x(.)$ by choosing a discrete set $DT = \{t_0 < t_1 < \ldots\}$ satisfying (c), defining $M(s, x, t) = A(s, x)$ and $H(s, x, t) = B(s, x)$ for $t \in DT$, and defining $M(s, x, t) = s$ and $H(s, x, t) = \textit{no action}$ for $t \notin DT$, where A and B are transition tables which are not dependent on time.

3.3 Automata as Strategies

Next we explain how we can use a continuous-input discrete-output automaton as a strategy for Control in our continuous sensing games. First choose an input alphabet $I = X$ and an alphabet of essential outputs $J = C \times [0, \infty)$. Control uses the continuous-input discrete-output automaton in the following way. Suppose t is the current time and $t_k < t$ is the last time the automaton output was an essential Control move. If $y(t)$ is the current input, the automaton stays in its current state s or shifts into another state according to its transition function $M(s, y(t))$ and outputs the respective Control move according to its output function $H(s, y(t))$. If at time t, there is a shift to another state, then the next essential Control move occurs at t and $t_{k+1} = t$. However the sequence of automaton states resulting may not form a run, much less a realizable run, in the course of reading an input. We call the automaton a *realizable strategy* for Control if whenever Control uses the automaton as its strategy, then the resulting play produces a realizable run.

Since the set of plant states is usually a subset of a Euclidean space, it is natural to consider automata with closed and separated switching sets as strategies for Control. However even if Control use this type of automaton, it will not always produce plays whose realizability can not be established by appealing to Theorem 7. The reason is that the automaton output affects the future input and may result in the automaton

input not being a continuous function of time, so that Theorem 7 does not apply. Such an automaton is given below.

Control strategies need not produce realizable runs

Consider the plant with a scalar control and disturbance:

$\dot{y} = d$, for $u = 0$
$\dot{y} = -d$, for $u = 1$
$d \in Z,\ d > 0$
$y(0) = 0$, and initially the control parameter is set to 0.

Consider the following automaton represented strategy for the player Control:

$S = \{0, 1\}$
$s_{in} = 0$
$I = \{y : y \in R\}$
$J = \{u := 0,\ u := 1\}$. (Here we think of the essential control moves as orders to set the control parameter to the indicated values.)

Let $\alpha > \beta > 0$ be given and let the transition table be defined by:

$$M(0, y) = \begin{cases} 1 \text{ if } y \geq \alpha \\ 0 \text{ if } y < \alpha \end{cases}$$

$$M(1, y) = \begin{cases} 0 \text{ if } y \leq \beta \\ 1 \text{ if } y < \beta \end{cases}$$

Here we assume The output function is defined by

$$H(0, y) = \begin{cases} u := 1 & \text{if } y \geq \alpha \\ \textit{no action} & \text{if } y < \alpha \end{cases}$$

$$H(1, y) = \begin{cases} u := 0 & \text{if } y \geq \beta \\ \textit{no action} & \text{if } y < \beta \end{cases}$$

It is easy to see that there are exactly two switching sets, namely, $G_0 = (\alpha, \infty)$ and $G_1 = (-\infty, \beta)$. Thus since $\beta < \alpha$, these are separated switching sets. If this strategy always produced realizable runs, then the corresponding plant trajectories would be continuous. But we exhibit a plant trajectory from a game which uses the automaton as a Control strategy and which produces a discontinuous plant trajectory. Assume that the disturbance is initially $d = 1$ at time $t = 0$, and that the disturbance doubles after each automaton state switch. We will get the first switch at time α, the second will occur $(\alpha - \beta)/2$ seconds later, the third will occur $(\alpha - \beta)/2^2$ after the second, and so on. The switch times t_k are the sums of the first k terms of this series. That is, they are $t_k = \sum_{i=0}^{k-1} \frac{\alpha - \beta}{2^i}$. this sequence has a finite limit point 2α. Hence there are times arbitrary close to 2α from the left where the plant state is $y = \alpha$ and the plant state $y = \beta$. Thus the plant trajectory is not continuous at $t = 2\alpha$.

Plants with Realizable Control Strategies

Next we define a class of plants together with a class of input-output automata which are guaranteed to produce realizable strategies for Control.

Suppose the plant is modeled by a system of differential equations of the form

$$\dot{x} = f(x, u, d), u \in U, d \in D, x \in X,$$

where u is a control parameter and d is disturbance parameter. Assume that $U \subseteq E^m$, $D \subseteq E^k$, and that $X \subset E^n$. We allow an additional source of nondeterminism in the plant of the following sort. For each pair of parameters (u, u'), there is a delay in resetting u to u', bounded by distance $\rho(u, u')$. Let $t^\star$ and $x^\star$ be the initial conditions for the plant. Assume that we have only piecewise constant control functions, so that we can identify control parameter value u with the constant control function $u(t) = u$ for all $t \geq t'$, where t' is a resetting time of a previous control parameter value to u.

Consider the following strategy A for Control in the continuous sensing game in which the plant state x is being sensed by Control. Let A be a continuous-input discrete-output automaton such that:

1. Its state space S is finite.
2. Its input alphabet is $I = X$.
3. Its alphabet of essential control moves is $J = U \times [0, \infty)$.
4. Its transition table M satisfies the condition of separateness of switching sets.
5. The switching set for the initial state contains the initial plant state $x^\star \in G_{s_{in}}$.
6. The automaton output function $H(s, x)$ produces an essential output only when x is in the switching set G_s, otherwise $H(s, x) = \textit{no action}$.

In particular, $H(s_{in}, x^\star) = (u, t^\star)$ for some $u \in U$ so that the first output to be produced by the automaton is an essential move.

Theorem 10. *Suppose that the plant is modeled by*

$$\dot{y} = f(t, y, u, d)$$

as described above. Suppose that f satisfies the Caratheodory conditions **CC 1, CC 2,** *and* **CC 3,** *where we assume that there is a fixed function $m(.)$ for f, independent of the value u of the control parameter for condition* **CC3.** *Assume that the automaton A described above has closed separated switching sets. Then for every play μ consistent with Control following the strategy A, the resulting automaton state function is a realizable run of A.*

Proof. We show first that the use of A by the player Control results in the production of runs of A. By the conditions in the paragraph preceding the theorem, the first control move according to A is essential. The control laws here are constant functions. Suppose there are no more essential control moves. Since the control laws here are constant functions, an essential move creates continuous plant trajectories because of the three Caratheodory conditions satisfied by the plant model. It then follows by Theorem 10 that the corresponding run is realizable.

Now suppose a finite number of essential control moves were made from the beginning of the play. Consider a time $t^{\star\star}$ at which the last essential control move was made. That is, if $s^\star$ is the state of A at time $t^{\star\star}$, then the automaton input $x^{\star\star}$ at time $t^{\star\star}$ is in the switching set $G_{s^\star}$. We may assume that the interaction of the automaton A and the plant has produced the plant trajectory γ up to time $t^{\star\star}$ and

$\gamma(t^{**}) = x^{**}$. In other words $\gamma(t^{**}) \in G_{s^*}$. We wish to show that there is a positive $\tau > 0$ such that there is no essential automaton outputs and transitions to new states in the interval $(t^{**}, t^{**} + \tau)$. It is easy to see that, even with the finite delay corresponding to resetting the control according to $\langle u, t^{**} \rangle = H(s^*, x^{**})$, the input to the automaton is a continuous plant trajectory because the function f satisfies the Caratheodory conditions **CC 1**, **CC 2**, and **CC 3**. It follows from Proposition 4.3 that there are two alternatives. The first alternative is that the automaton stays forever in this state. This produces a valid realizable run and play with t^{**} being the last essential Control move. The second alternative is that there is $\tau > 0$ such that the next essential move occurs at $t^{**} + \tau$. this means that there are no essential outputs or transitions to new states in $(t^{**}, t^{**} + \tau)$, since these occur at the same time according to the definition of A, and the transitions to new states do not occur in this alternative for such an interval.

It follows that when Control uses A in the game, the set of times at which essential moves are made is a discrete set. We let $DT(\mu)$ denote this set for the play μ. Next fix a play μ and consider an initial sequence $\{t_k : k \geq 0\}$ of $DT = DT(\mu)$ that begins with the time of the first essential control move. Clearly, every finite initial segment of this sequence determines a position in the game for which there is a corresponding continuous plant trajectory. Suppose that q is a limit point of this sequence. Due to assumption **CC 3**, we get the following estimate for any plant trajectory γ:

$$|\gamma(t') - \gamma(t)| \leq \int_t^{t'} m(t)dt.$$

Since the integral is absolutely continuous, it follows that the

$$lim_{k\to\infty}|\gamma(t_{k+1}) - \gamma(t_k)| = 0.$$

By the argument above, it follows that for every k, $\gamma(t_k) \in G_{s_k}$, where s_k is the automaton state at which the transition occurs at the switch time t_k. Moreover, the sets $G_{s_{k+1}}$ and G_{s_k} are distinct since the states s_{k+1} and s_k are distinct. The fact that the above limit is 0 and the fact that there are only a finite number of switching sets would imply that for some pair of states s and s', the distance between G_s and $G_{s'}$ is 0. But this contradicts our assumption that there is nonzero distance between the switching sets. Thus the sequence $\{t_k\}$ has no finite limit points. Since the set of switching instants DT is the set of times of essential Control moves and this set has no finite limit points, the corresponding play is realizable. □

We note that for the water level monitor problem describe above, the bound function for the Caratheodory condition $CC\ 3^*$ is $m(t) = max\{a, b\} \cdot t$. Thus the plays produced by the suggested automaton controller are all realizable.

4 From Continuous to Discrete Sensing Games

In this section we fully analyze the water pump example. We start by explicitly constructing a continuous-input discrete-output automaton $A(g, h)$ for a pair of parameters $g > h$ as described in the previous section. By Theorem 10, we know that if Control uses $A(g, h)$ for its strategy in the continuous sensing game for the water

level monitor, then it will always produce realizable runs for $A(g,h)$. We shall show that for any desired water levels, $u < v$, we can pick (g,h) in such a way that if Control uses the automata $A(g,h)$ for its strategy in the continuous sensing game, then Control will win in the sense that we will guarantee that at all times t, the water level $y(t)$ will satisfy $u \leq y(t) \leq v$ assuming that $u \leq y(0) \leq v$. Then we shall show how we can use the continuous-input discrete-output automaton $A(g,h)$ to design a finite automaton which will control the plant, that is, the water tank plus pump, to meet the desired performance specification. Finally, we shall show that we can explicitly extract Kohn-Nerode small topologies which will verify the controllability and observability of our discrete control strategy.

The (g,h)-Automaton $A(g,h)$

With any pair (g,h) of positive numbers with $g > h$ we associate a (g,h)-automaton $A(g,h)$ with continuous input alphabet and a three letter output alphabet. (This is essentially the same automaton that was described in the previous section.)

1. The input alphabet consists of the numbers in interval M of possible water levels y.
2. The two automaton states are $son, soff$.
3. The three letter output alphabet is $pon, poff, no\ action$.

The transition table and output function of this automaton are defined as follows.

$$M(son,y) = \begin{cases} soff & \text{if } y \geq g \\ son & \text{if } y < g \end{cases}$$

$$M(soff,y) = \begin{cases} son & \text{if } y \leq h \\ soff & \text{if } y > h \end{cases}$$

$$H(son,y) = \begin{cases} soff & \text{if } y \geq g \\ no\ action & \text{if } y < g \end{cases}$$

$$H(soff,y) = \begin{cases} son & \text{if } y \leq h \\ no\ action & \text{if } y > h \end{cases}$$

The the switching sets for $A(g,h)$ are $G_{son} = [g,\infty)$ and $G_{soff} = (-\infty,h]$. We can thus guarantee that $A(g,h)$ has separated switching sets if we impose the requirement that $g > h$.

Theorem 11. *If parameters g and h in the continuous sensing game for the water level monitor with maximum delay d satisfy the conditions:*

$$\begin{array}{lll} (1)\ g > h, & (2)\ g + a \cdot d \leq v, & (3)\ h - b \cdot d \geq u, \\ (4)\ h - b \cdot d \leq y(0) < g, & (5)\ \frac{g-h}{a} > d, & (6)\ \frac{g-h}{b} > d, \end{array} \tag{2}$$

then the strategy $A(g,h)$ is a winning strategy for the player Control in any game where the initial state of Control is son and the initial state of the pump is pon.

Proof Suppose $\mu = Y_0, z_0, Y_1, z_1, \ldots$ is a play consistent with A. We have to show two things. First we must show that the strategy induced by $A(g,h)$ is applicable at every position of Control in this play. That is, we must show that if Control using this strategy, then he never gets stuck in the sense that he is unable to make a move according to the strategy. This is the perpetual property [48], [38]. Second we must show that $A(g,h)$ induces a winning strategy for Control, i.e. that the water level trajectory $y(.)$ corresponding to any play consistent with the game initial condition and the strategy $A(g,h)$ has the property that for all times $t \geq 0$, $u \leq y(t) \leq v$. We show both properties by induction on the length of a position in the play.

The initial position of the play is $p = \langle pon, 0 \rangle$ and the initial trajectory of the plant is just $\langle 0, y(0) \rangle$. That is, the initial control sent to the plant is that the pump should be on. Now since $y(0)$ satisfies $v \leq h - b \cdot d \leq y(0) \leq g < v$, we see that the initial trajectory is within acceptable bounds.

Consider the first block of plant moves which is specified by its corresponding trajectory Y_0. By our assumptions, we have that for all t,

$$0 < a' \leq \dot{Y_0}(t) \leq a.$$

Thus Y_0 will be a strictly increasing function so that there will be some time $t_1 > 0$ such that $Y_0(t_1) = g$. It is easy to see that $t_1 \leq (g - y(0))/a'$

Thus at time t_1, Control issues the order that the pump should be turned off and switches to state $soff$. Thus $z_0 = \langle poff, t_1 \rangle$.

Now consider the next block of plant moves which is specified by its corresponding trajectory Y_1. Because of the delay in switching from the pump being on to the pump being off after the control order to turn the pump off has been issued, there is some $0 \leq \tau_1 \leq d$ such that the pump remains on between time t_1 and time $t_1 + \tau_1$ and then the pump turns off. Thus the corresponding trajectory Y_1 satisfies

$$\begin{array}{ll} 0 < a' \leq \dot{Y_1}(t) \leq a & \text{if } t_1 \leq t \leq t_1 + \tau_1 \\ 0 > -b' \geq \dot{Y_1}(t) \geq -b & \text{if } t > t_1 + \tau_1 \end{array}$$

It is then easy to see that the trajectory Y_1 must reach its maximum at time $t = t_1 + \tau_1$ and that this maximum value is bounded by $g + a\tau_1 \leq g + ad \leq v$. After time $t_1 + \tau_1$, Y_1 is strictly decreasing so that there must be some time $t_2 > t_1$ such that $Y_1(t_2) = h$. It is easy to see that $d < \frac{g-h}{b} \leq t_2 - t_1 \leq \frac{g+ad-h}{b'}$. Since $t_2 - t_1 > d$ it follows that the state the pump will be $soff$ at time t_2. Thus at time t_2, Control issues a order that the pump be turned on and switches to state son. Thus $z_1 = \langle pon, t_2 \rangle$. It then easily follows that the values of the trajectory Y_1 between times $t = t_1$ and $t = t_2$ takes on its maximum value at time $t_1 + \tau_1$ and its minimum value at time t_2 where $Y_1(t_2) = h$. Thus the values of $Y_1(t)$ lie between h and $g + ad$ and hence meets our performance specifications.

Now consider the next block of plant moves which is specified by its corresponding trajectory Y_2. Again, because of the delay in switching from the pump being off to the pump being on after the control order to turn the pump on has been issued, there is some $0 \leq \tau_2 \leq d$ such that the pump remains off between time t_2 and time $t_2 + \tau_2$ and then the pump turns on. Thus the corresponding trajectory Y_2 satisfies

$$\begin{array}{ll} 0 > -b' \geq \dot{Y_2}(t) \geq -b & \text{if } t_2 \leq t \leq t_2 + \tau_2 \\ 0 < a' \leq \dot{Y_2}(t) \leq a & \text{if } t > t_2 + \tau_2 \end{array}$$

It is then easy to see that the trajectory Y_2 must reach its minimum at time $t = t_2+\tau_2$ and that this minimum value is bounded below by $h - a\tau_2 \geq h - bd \geq u$. After time $t_2 + \tau_2$, Y_2 is strictly increasing so that there must be some time $t_3 > t_2$ such that $Y_2(t_3) = g$. It is easy to see that $d < \frac{g-h}{a} \leq t_3 - t_2 \leq \frac{g-h+bd}{a'}$. Since $t_3 - t_2 > d$ it follows that the state the pump will be *son* at time t_3. At time t_3, Control issues a order that the pump be turned off and switches to state $soff$. Thus $z_2 = \langle poff, t_3 \rangle$. It then easily follows that the values of the trajectory Y_2 between times $t = t_2$ and $t = t_3$ takes on its minimum value at time $t_2 + \tau_2$ and its maximum value at time t_3 where $Y_2(t_3) = g$. Thus the values of $Y_2(t)$ lie between $h - bd$ and g and hence meet our performance specifications.

Thus the behavior of the system between the position ending in z_1 and the position ending in z_3 meets the performance specification and the requirement that Control can follow the strategy determined by $A(g, h)$. Note that at time t_1, the water level is g and the pump is on and at time t_3 the water and the pump is on. It is then straightforward to prove by induction that at time t_{2n+1} the water level will be g and the pump will be on and that exactly the same analysis will apply to the behavior of the system between the position ending in z_{2n+1} and the position ending in z_{2n+3}. Hence it follows that the strategy for Control induced by $A(g, h)$ is a winning strategy for Control as claimed. □

It should be clear that in the statement of Theorem 11 we can replace the assumption that the pump is initially on and $u \leq y(0) < g$ by the assumption that the pump is initially off and $h < y(0) \leq g + a \cdot d$ and the conclusion of the Proposition will continue to hold.

4.1 The $(g, h)-$Automata for Discrete Sampling and Measurement Errors

We now modify our continuous sensing game for the water level monitor in two ways. First we shall assume that Control, instead of continuously sensing the plant state, senses the plants state only at discrete times $t_0 < t_1 < t_2 < \ldots$, where there is some positive $\Delta > 0$ such that $t_{k+1} - t_k \geq \Delta$ for all $k \geq 0$.

Second, we shall assume that Control is not able to exactly measure the plant state, but only that Control can measure the plant state within some error e. Our goal is to specify a continuous-input discrete-ouput automaton strategy for Control in such a game and the sequence of sampling times $t_0 < t_1 < t_2 < \ldots$ so that if Control measures the plants state at the times $t_0 < t_1 < t_2 < \ldots$ with an error of no more than e and follows the strategy induced by the continuous-input discrete-ouput automaton, then Control will ensure that the plant meets the performance specifications.

In this case, we shall assume that $t_0 = 0$ and that $t_k = k\Delta$ for all $k > 0$ so that we are sampling every Δ seconds, where $\Delta > d$ and d is the maximum delay that can occur between the time at which Control issues a order to the pump to turn off or on and the time the pump actually achieves the state required by the order. Moreover, we shall continue to use the automaton $A(g, h)$ for the strategy for Control. Thus the behavior of the system is the following:

A. Suppose that the automaton is in state *son* and receives as input measurement m. Then, instantaneously,

1. if $m \geq g$, then the automaton outputs $poff$ and also shifts its state to $soff$, and
2. if $m < g$, then the automaton remains in state pon, and outputs *no action*.

B. Suppose that the automaton is in state $soff$ and and receives input measurement m. Then, instantaneously,

1. if $m \leq h$, then the automaton outputs pon and shifts to state son, and
2. if $y > h$, then the automaton remains in state $soff$ and outputs *no action*.

Thus our problem is find Δ and the parameter g and h to ensure that the water level $y(t)$ stays within the desired bounds, i.e. that for all t, $u \leq y(t) \leq v$. First of all, since we pick $\Delta > d$, we will be guaranteed that the plant and automaton states correspond to each other at the end of each sampling interval. That is, if initially the plant state and the initial state of $A(g,h)$ are such that if the initial state of $A(g,h)$ is $soff$, then the pump is off and if initial state of $A(g,h)$ is son, then the pump is on, then at some time before the end of each sampling interval the state of $A(g,h)$ the pump will correspond to each other.

It is then quite easy to derive the necessary conditions on the parameters g and h to guarantee that the control automaton $A(g,h)$ provides a winning strategy for Control in our modified game. That is, all we do have to do is analyze the plant trajectories for given input measurement and states of $A(g,h)$. We consider the following cases.

Case 1 Suppose that the plants state is son and at time t_k, Control receives a measurement $m_k < g$. Now by assumption, if the actual water level at time t_k is $y(t_k)$, then

$$m_k - e \leq y(t_k) \leq m_k + e.$$

Assume also that the pump is on at time t_k so that in this case the automaton remains in state son and issues the order *no action* and the pump remains on for the next Δ seconds. Then since the plant trajectory $y(.)$ between t_k and $t_{k+1} = t_k + \Delta$ must satisfy

$$0 < a' \leq \dot{y}(t) \leq a,$$

it is easy to see that $y(t)$ is a strictly increasing function in this interval and that

$$y(t_{k+1}) \leq y(t_k) + a\Delta \leq m_k + a\Delta + e \leq g + a\Delta + e.$$

Now if we find that the measurement received at time t_{k+1}, m_{k+1}, is still less than g, then of course the automaton will continue to be in state son and issue the order *no action* so that the pump will remain on, the plant trajectory $y(.)$ between t_{k+1} and t_{k+2} will be strictly increasing, and $y(t_{k+2}) \leq g + a\Delta + e$. We will continue on this way until we find the least $l > k$ such that the measurement received at time t_l will be greater than or equal to g. By our analysis, the actual plant state $y(t_l)$ will bounded by $g + a\Delta + e$. At that point, the automaton will issue the order for the pump to be turned off and switch to state $soff$. What happens to the trajectory $y(t)$ between times t_l and $t_{l+1} = t_l + \Delta$? It is easy to see that our analysis of Theorem 11 now applies. That is, there will be some $\tau_l \leq d < \Delta$ such that the trajectory satisfies

$$\begin{aligned} 0 < a' \leq \dot{y}(t) \leq a \quad & \text{if } t_l \leq t \leq t_l + \tau_l \\ 0 > -b' \geq \dot{y}(t) \geq -b \quad & \text{if } t_l + \tau_l < t \leq t_{l+1}. \end{aligned}$$

It is then easy to see that the trajectory $y(t)$ in the interval $[t_l, t_{l+1}]$ must reach its maximum at time $t = t_l + \tau_l$ and that this maximum value is bounded by $y(t_l) + a\tau_1 \leq y(t_l) + ad \leq g + a\Delta + e + ad$. Then after time $t_l + \tau_l$, $y(t)$ is strictly decreasing. It is now easy to see that if we pick g so that

$$g + ad + a\Delta + e \leq v,$$

then we will ensure that following the $A(g,h)$ strategy will ensure that the water level never becomes greater than v. There is also a lower bound which is imposed on g which comes from the fact that the minimum value of $y(t)$ in the interval $[t_l, t_{l+1}]$ must be greater than or equal to u. Since we are assuming that $m_l \geq g$, we know that $y(t_l) \geq g - e$. If we assume that there is no delay in turning the pump off, then $y(t)$ could be strictly decreasing in the interval. It is then easy to see that in such a situation, $y(t_{l+1})$ could be as small as $g - e - b\Delta$. Moreover it could be that $g - e - b\Delta - e \leq h$ so that $m_{l+1} \leq h$. In that situation, the pump will be off and our controller would tell the pump to turn on. However there could be a maximum delay of time d before the pump turns on and the the water level once again starts to increase. Thus there could be a further drop of $-bd$ in the water level during this delay so that the water level could become as small as $g - e - b\Delta - bd$. Thus we must also assume that $g - bd - b\Delta - e \geq u$ or equivalently that $u - bd + b\Delta + e \leq g$. In case 2, we will deal with the case when $m_{l+1} > h$.

Case 2. Suppose that the plants state is $soff$ and at time t_k, Control receives a measurement $m_k > h$. Again the actual water level $y(t_k)$ satisfies

$$m_k - e \leq y(t_k) \leq m_k + e.$$

Assume also that the pump is off at time t_k so that in this case the automaton remains in state $soff$ and issues the order *no action* and the pump remains off for the next Δ seconds. Then since the plant trajectory $y(.)$ between t_k and $t_{k+1} = t_k + \Delta$ must satisfy

$$0 > -b' \geq \dot{y}(t) \geq -b,$$

it is easy to see that $y(t)$ is a strictly decreasing function in this interval and that

$$y(t_{k+1}) \geq y(t_k) - b\Delta \geq m_k - b\Delta - e \geq h - b\Delta - e.$$

Now if we find that the measurement received at time t_{k+1}, m_{k+1}, is still greater than h, then of course the automaton will continue to be in state $soff$ and issue the order *no action* so that the pump will remain off, the plant trajectory $y(.)$ between t_{k+1} and t_{k+2} will be strictly decreasing, and $y(t_{k+2}) \geq h - b\Delta - e$. We will continue on this way until we find the least $l > k$ such that the measurement received at time t_l will be less than or equal to h. By our analysis, the actual plant state $y(t_l)$ will bounded below by $h - b\Delta - e$. At that point, the automaton will issue the order for the pump to be turned on and switch to state *son*. Again use our analysis of Theorem 11 to analyze what happens to the trajectory $y(t)$ between times t_l and $t_{l+1} = t_l + \Delta$. That is, there will be some $\tau_l \leq d < \Delta$ such that the trajectory satisfies

$$\begin{aligned} &0 > -b' \geq \dot{y}(t) \geq -b \quad \text{if } t_l \leq t \leq t_l + \tau_l \\ &0 < a' \leq \dot{y}(t) \leq a \qquad \text{if } t_l + \tau_l < t \leq t_{l+1}. \end{aligned}$$

It is then easy to see that the trajectory $y(t)$ in the interval $[t_l, t_{l+1}]$ must reach its minimum at time $t = t_l + \tau_l$ and that this minimum value is bounded below by $y(t_l) - b\tau_1 \geq y(t_l) - bd \geq h - b\Delta - e - bd$. Then after time $t_l + \tau_l$, $y(t)$ is strictly increasing. It is now easy to see that if we pick h so that

$$h - bd - b\Delta - e \geq u,$$

then we will ensure that following the $A(g,h)$ strategy will ensure that the water level never becomes less than u. There is also upper bound which is imposed on h which comes from the fact that the maximum value of $y(t)$ in the interval $[t_l, t_{l+1}]$ must be less than or equal to v. Since we are assuming that $m_l \leq h$, we know that $y(t_l) \leq h + e$. If we assume that there is no delay in turning the pump on, then $y(t)$ could be strictly increasing in the interval. It is then easy to see that in such a situation, $y(t_{l+1})$ could be as large as $h+e+ba\Delta$. Note that the case when $m_{l+1} < g$ was handled in Case 1. However it could be that $h+e+a\Delta+e \geq g$ so that $m_{l+1} \geq g$. In that situation, the pump will be on and our controller would tell the pump to turn off. However there could be a maximum delay of time d before the pump turns off and the the water level once again starts to decrease. Thus there could be a further rise of ad in the water level during this delay so that the water level could become as large as $h + e + a\Delta + ad$. Thus we must also assume that $h + ad + a\Delta + e \leq v$ or equivalently that $h \leq v - ad - a\Delta - e$.

Below is the proposition asserting the conditions for correctness of the $A(g,h)$ control automaton.

Theorem 12. *Suppose in the discrete sampling game for the water level monitor, we have a maximum delay of d for switching plant states, we are given a finite sampling time $\Delta > d > 0$ and a measurement error bound $e \geq 0$. Choose the numbers $h < g$ so that*

$$u + bd + b\Delta + e \leq g, h \leq v - ad - a\Delta - e.$$

Suppose that the initial water level is between $h + e$ and $v - a \cdot d$ and the pump is on or the initial water level is between $u + b \cdot d$ and $g - e$ and the pump is off. Suppose that initially the pump and the control automaton are both in the "on" state or both in the "off" state. With the $A(g,h)$-controller introduced above, the water level satisfies the performance specification that $u \leq y(t) \leq v$ at all times $t \geq 0$.

Proof By using the analysis of Case 1 and Case 2 above, one can easily prove by induction k that if Control follows the $A(g,h)$ strategy in our modified game, then in each interval $[t_k, t_{k+1}]$, the trajectory of the plant $y(t)$ will always satisfy that $u \leq y(t) \leq v$. We leave the details to the reader. □

We note that the inequalities on g and h in Theorem 12 automatically impose the following upper bound on the size of the sampling interval Δ:

$$\Delta \leq \frac{v - u + d(a+b) - 2e}{a+b}.$$

4.2 Topological Finite Automata from Open Covers

In appendix II of [24], there is a general method which, given a hybrid system whose performance specification is autonomous, extracts a finite automaton which which can be used to guarantee that the hybrid system will meet its performance specifications as well as to extract small topologies which guarantee the stability of the system. Our goal in this section is to follow appendix II of [24] and construct a finite open cover yielding a finite control automaton and small topologies for our water level monitor example which guarantee that the water level always stay within specified bounds. Here, when we say that the performance specification is autonomous, we mean the following. We assume that the plant is modeled by a differential equation

$$\dot{y} = f(y, u, d)$$

where u is a control parameter and d is a disturbance parameter. Then in each interval of time $\Delta = [t_0, t_1]$ and any given plant state y that lies within a certain set of acceptable values, we want to find a control law $u(.)$ such that if we use the control law $u(t)$ to determine the plant trajectory, then for any acceptable disturbance $d(t)$, our plant trajectory should meet the required performance specification. That is, any function $y(t)$ such that $y(t_0) = y$ and $\dot{y}(t) = f(y(t), u(t), d(t))$ for all $t \in \Delta$ must meet our performance specification. We assume that our choice of suitable control functions $u(t)$ for any interval Δ depends only on the plant state x and the internal state of the controller but not on the time t which is the start of the interval. In this situation, the problem of meeting performance specification is equivalent to determining a set Q of "acceptable "pairs $(x, u(t))$ of plants states and control functions. That is, each pair represents a plant trajectory which begins at the plant state of the pair and is guided by the control law of the pair which satisfies the our performance specifications over the sampling interval Δ. Note that in this situation, the control law $u(t)$ is a function of time over the sampling interval that takes values in the range of values of the control parameter.

For example, the range of the control parameter for the water pump-tank system is the set of orders for the pump or equivalently the set of states of the pump $\{1 = 'pon', 0 = 'poff'\}$ and every control law is a constant function over the sampling interval with the range being the pump states. In what follows, we adapt the definition of the set of pairs Q to reflect the presence of possible delays in switching the pump states. Thus for the water tank and pump example, we let Q consist of the pairs such that for any admissible delay in switching to a new pump state as directed by Control in the sampling interval Δ, the the water level which correspond to the plant trajectory stays within our required bounds.

In the general setting for autonomous performance specification, the first stage of finding a control automaton in the small topologies satisfying the specification is to find a control function.

Definition 13. A **feedback control function** H is a map that assigns to each pair of a plant state x reached at the end of a sampling interval Δ and the current control law u used in Δ, a control law u' such that the corresponding plant trajectory over the next sampling interval Δ' satisfies the performance specification over that interval.

A useful model to keep in mind is to think of the control $u(t)$ as being determined by a physical controller. Thus the automaton communicates with the physical controller by setting the state of the physical controller s_u which has the effect of imposing the control $u(t)$ for the next sampling interval. In such a situation, we can identify the control laws with the states of the physical controller. For example, in the case of the water pump and tank example where the control functions are piecewise constant, we may represent u its value which is either pon or $poff$. For the rest of this section, we shall use this model so that instead of talking about the current control law of the sampling interval, we will talk about the current state of the physical controller, etc.

Definition 14. The automaton $A(H)$ associated with a control function H is defined as follows.

1. Its set of states is the set of states of the physical controller K. (In the more general language, K would be the set of possible control laws which occur in pairs in Q.)
2. Its input alphabet is the set of plant states $U = PS$.
3. Its output function $H(u, k)$ is the feedback control function.
4. Its transition table $M(u, k)$ models the switching of control laws output by the controller, i.e. $M(u, k) = H(u, k)$ for all $u \in U$ and $k \in K$.

Next we want to isolate some properties of the automaton $A(H)$ or equivalently the feedback control function H which will guarantee that we can perpetually apply our control strategy.

Definition 15. Say that the automaton $A(H)$ associated with a feedback control function H is **correct with respect a performance specification and a region** $B \subseteq PS \times K$ if the following holds. For any pair $a = (y, k) \in B$ and for any admissible disturbance $d(t)$, any trajectory beginning from y and guided by the control corresponding to k during the delay for switching to new state of the physical controller $H(y, k)$ and by the control $H(y, k)$ after the delay satisfies the performance specifications and ends up in B at the end of the sampling interval. Here "ends up in B" means that if y_1 is plant state corresponding the trajectory at the end of the sampling interval, then $(y_1, H(y, k)) \in B$.

Definition 16. Suppose that there is a region B in the domain of the feedback control function H such that for any pair $(y, k) \in B$ and for any admissible disturbance $d(t)$, any trajectory beginning from y and guided by the control corresponding to k during the delay for switching to new state of the physical controller $H(y, k)$ and by the control control $H(y, k)$ after the delay satisfies the performance specifications and ends up in B at the end of the sampling interval. Then we call such a control function a **guiding feedback control function** relative to B.

The definitions above can easily be extended to apply to the case when the control function is set-valued as introduced in appendix II of [24]. The idea of a set-valued feedback control function is that one computes a set of controls or in our case a set of physical controller states from a pair consisting of a plant state and a physical

controller state and then selects from that set one control or physical controller state which will be used to determine the plant trajectory in the next sampling interval Δ. The set of control functions or physical controller states that we compute should be such that for every control function or physical controller state that could have been chosen from the set and every admissible disturbance, the corresponding plant trajectory always satisfies the performance specifications.

Set-valued feedback control functions arise naturally in our context. Consider a map H from the pairs (m, k) ((measurement, physical controller state)) into the set of states of the physical controller. If we take the measurements as inputs to the control automaton and identify the map H with the control automaton output function, we of course have an ordinary function as opposed to a set-valued one. However suppose that we assume that a measurement can be any value that approximates a plant state within some error bound. That is, we view a measurement as a set-valued function over plant states from which an nondeterministic choice of an element from a set is made. For example, suppose that the map above is $H(m, k)$, where m is a measurement, and k is a physical controller state. Then the corresponding set-valued feedback control function is $G(y, k) = \{H(m, k) : |m - y| \leq e\}$. Here $e > 0$ is the measurement error bound.

In appendix II [24], the graph of G is assumed closed. But our G is not closed. So we take the closure of the graph of G and consider a corresponding set-valued function G' whose graph is that closure. So our control function will be G'. The topologies that are used in the construction of G' are the natural Hausdorff topologies on the plant state space and on the space of states of the physical controller following [24]. The fact that the topological spaces are Hausdorff means that if the state space K of the physical controller is finite, then the K must have the discrete topology since the only Huassdorff topology on a finite set is the discrete topology.

It is also important for applying the methodology of appendix II that the domain of the feedback control function be a subset of the set Q. This is true of the graph of G but not necessarily for the closure of G because the domain of the closure of G may include boundary points of Q which are not in Q. In the case we consider, the closure of G will in fact lie entirely in Q.

Now let us go back to our water level monitor example. Let $K = \{pon, poff\}$ be the range of control values or equivalently the states of the pump. Let the variable k range over the set K. Here, the map $H(m, k)$ is defined by

$$H(pon, y) = \begin{cases} poff \text{ if } y \geq g \\ pon \quad \text{otherwise} \end{cases}$$

$$H(poff, y) = \begin{cases} pon \quad \text{if } y \leq h \\ poff \text{ otherwise} \end{cases}$$

A water level y is taken from the set $[u, v]$, which carries the natural Euclidean topology. There is only one Hausdorff topology on the set K, the discrete topology.

To construct the function G note that for each control automaton state the function H is continuous except at one point in the range of y. The point of discontinuity for H is either g or h at respective automaton states on, off. It follows that if y is separated from, say g, by more than the error bound e, then the function has a singleton set as a value. One can see that at points $g-e, g+e, h-e, h+e$, the value

of G is still a singleton. At points near to g, h by less than e, the value of G is K since H has a different value to the right of g than to the left of g. Thus

$$G(pon,y) = \begin{cases} \{pon\} & \text{if } y < g-e \\ K & \text{if } g-e \leq y < g+e \\ \{poff\} & \text{if } y \geq g+e \end{cases}$$

$$G(poff,y) = \begin{cases} \{pon\} & \text{if } y \leq h-e \\ K & \text{if } h-e < y \leq h+e \\ \{poff\} & \text{if } y > h+e \end{cases}$$

Now consider the closure G' of the graph of G. Here we use the same letter for the set-valued function and for its graph. Here is the resulting closure.

$$G'(pon,y) = \begin{cases} \{pon\} & \text{if } y < g-e \\ K & \text{if } g-e \leq y \leq g+e \\ \{poff\} & \text{if } y > g+e \end{cases}$$

$$G'(poff,y) = \begin{cases} \{pon\} & \text{if } y < h-e \\ K & \text{if } h-e \leq y \leq h+e \\ \{poff\} & \text{if } y > h+e \end{cases}$$

Note that the definition of the function G also makes sense for exact measurements ($e = 0$), but in that case the corresponding function G' is multi-valued only at the points of discontinuity $(g,on), (h,off)$ of H. This nondeterminacy makes clear the arbitrary nature of the choice of a strict or non strict inequality in the definition of H. That is, we obtain four functions which are variants of H, differing from H only in having non-strict inequalities in the definition. All give rise to the same G'.

We distinguish between three slightly different automata, Aut_1, Aut_2, and Aut_3, which depend on our pair of parameters g and h. For all three automaton, the set of states is $\{pon, poff\}$, the input alphabet is the set of water levels and the output alphabet the same as the set of states. Thus we need only define their output functions $H_i(y,k)$ and their transition tables $M_i(y,k)$. For the automaton $Aut_1(g,h)$, the output function $H_1(y,k)$ and the transition table $M_1(y,k)$ are both equal to the function $H(y,k)$ defined above. If we think of this automaton as a strategy for Control in the discrete sampling game with error measurements, then Aut_1 gives essentially the same strategy as the automaton $A(g,h)$ described in the previous section. The only difference between the two automaton is when in the state *pon* when $y \leq g$, $Aut_1(g,h)$ outputs *pon* while $A(g,h)$ outputs *no action*. However we regard both of these instructions to a pump which is on to be the same, i.e. they both keep the pump on. Similarly when in the state $poff$ when $y \geq h$, $Aut_1(g,h)$ outputs $poff$ while $A(g,h)$ outputs *no action*. Again we regard both of these instructions to a pump which is off to be the same, i.e. they both keep the pump off. Thus by Proposition 12 Aut_1 is a winning strategy for Control in the discrete sampling game with error measurements. Now as observed above, if we think about the action of the strategy as a function of plant states instead of on measurements where we assume that the absolute value of the difference between the measurement and the actual plant state is no more than e, then the transition table and the output

function are nondeterministic and are give by the function G defined above. Thus we define a second automaton $Aut_2(g,h)$ whose transition table and output function are given by G, i.e. for all (y,k), $M_2(y,k) = H_2(y,k) = G(y,k)$. Of course Aut_2 is a nondeterministic automaton and the output function is set valued. We shall assume that the automaton operates as follows. If Aut_2 is in state s and is reading input y and goes to state s' at its next step so that $s' \in M_2(s,y)$, then the output of the automaton in that circumstance is also s'. That is, our definitions ensure that for the pair (s,y), the possible new states and the possible outputs come from the same set since $M_2(s,y) = H_2(s,y)$. We are thus making the additional assumption that such choices are coordinated for any (s,y). In this way, we can use Aut_2 as a strategy for Control since our assumption will ensure that the internal state of the automaton Aut_1 and the state of pump are always coordinated at the end of sampling intervals if they start out coordinated. If we think of Aut_2 as a strategy for Control in the discrete sampling game without errors in measurements, i.e. in the discrete sampling game where the error bound $e = 0$, then this strategy for Control will produce exactly the same set of runs with respect to plant states as the strategy Aut_1 produces in the discrete sampling game with error measurements. Hence Aut_2 is a winning strategy for control in the discrete sampling games without error measurements. Finally we consider yet another nondeterministic automaton Aut_3 whose transition table and output function are given by G' instead of G. Again we assume that Aut_3 operates so that if Aut_3 is in state s and is reading input y and goes to state $s' \in M_3(s,y)$ at its next step, then the output of the automaton in that circumstance is also s'.

Remark

The differences between the control strategy Aut_1 in our discrete sampling game with errors in measurements bounded by e and the control strategy Aut_2 in our discrete sampling game without error measurements can be explained in terms whether we consider the analog to digital converter as part of the plant or whether we want to consider the analog to digital converter as part of the digital controller. That is, if we consider the analog to digital converter as part of the plant, then it is natural to assume that the digital controller receives only plant measurements and this situation is most naturally modeled as a discrete sampling game with errors in measurements where the control automaton is deterministic. However, if we consider the analog to digital converter as part of the digital controller, then the most natural way to model this situation is that we have a discrete sampling game without errors in measurements and that the control automaton behaves in a nondeterministic manner as described by Aut_2. Thus our choice of using Aut_1 in a discrete sampling game with errors in measurements or of using Aut_2 in a discrete sampling game without errors in measurements comes down to the choice of where in Figure 1 we wish to place the analog to digital converter, i.e on the digital side or on the analog side.

Our next proposition states that Aut_3 is also a winning strategy for Control in the discrete sampling games without error measurements.

Theorem 17. *Suppose in the discrete sampling game without errors in measurements for the water level monitor, we use finite sampling intervals of size Δ and that the maximum delay d for switching to new plant state is such that $\Delta > d > 0$. In addition assume $e \geq 0$ and that g and h satisfy*

1. $g \leq v - a \cdot d - a \cdot \Delta - e$;

2. $g - e \geq h + e$;
3. $h \geq u + b \cdot d + b \cdot \Delta + e$.

Suppose that the initial water level is between $h+e$ and $v-a \cdot d$ and the pump is on, or the initial water level is between $u+b \cdot d$ and $g-e$ and the pump is off. Suppose that initially the pump and any of the two control automata, Aut2 or Aut3, are both in the on state or both in the off state. Then Aut_2 and Aut_3 are winning strategies for Control in such discrete sampling games without errors in measurements for the water level monitor.

Proof The proof of Theorem 12 that $Aut1$ is a winning strategy for Control in discrete sampling games with errors in measurement bounded by e can be is easily adapted to prove that that either Aut_2 or $Aut3$ is a winning strategy for Control in the discrete sampling games without errors in measurement. The proof is by induction on the length of positions as before. We leave the details to the reader. □

The content of Theorem 17 can be restated as the following property of the feedback control function G'. Suppose the water level y is between $h+e$ and $v-a \cdot d$ and the pump is on or y is between $u+b \cdot d$ and $g-e$ and the pump is off. Suppose that the next control law is chosen from the set $G'(y,k)$, where k is the state of the pump as specified above at the beginning of the sampling interval Δ. Then the water level lies in the interval $[u,v]$ over the next sampling interval. Thus G' can indeed be used as a feedback control function for the water level and pump states in the region

$$A = [h+e, v-a \cdot d] \times \{on\} \cup [u+b \cdot d, g-e] \times \{off\}.$$

Moreover, the water level and the state of the pump at the end of the sampling interval satisfy the same assumptions that are satisfied by this data at the beginning of the sampling interval. That is, the trajectories that have begun in A will end in A at the end of a sampling interval if they are guided by a control law determined by the set-valued control function. According to our earlier definition, the feedback control function G' restricted to A is a guiding feedback control function.

Constructing Open Covers

We now consider an open cover of the graph of G' restricted to the region A. Our goal is to construct a finite automaton with small topologies approximating G'. We presented A above as a disjoint union of two open and closed (clopen) sets. Correspondingly, the graph of G' is a disjoint union of clopen sets. It is sufficient to cover each of the clopen sets independently. Choose $\epsilon > 0$ so small that the sets below are subsets of the graph of G'. To visualize the regions below more clearly, recall that we have the following the inequalities:

$$h - e \leq g - e \leq g + e \leq v - a \cdot d \text{ and}$$

$$u + b \cdot d \leq h - e \leq h + e \leq g - e.$$

Here is the open cover for the first clopen set:

$$V_1 = [h+e, g-e+\epsilon) \times \{on\} \times \{on\},$$
$$V_2 = (g-e-\epsilon, g+e+\epsilon) \times \{on\} \times K,$$
$$V_3\ (g+e-\epsilon, v-a \cdot d] \times \{on\} \times \{off\}.$$

Similarly here is an open cover for the second clopen set:

$$V_4 = [u + b \cdot d, h - e + \epsilon) \times \{off\} \times \{on\},$$
$$V_5 = (h - e - \epsilon, h + e + \epsilon) \times \{off\} \times K,$$
$$V_6 = (h + e - \epsilon, g + e] \times \{off\} \times \{off\}.$$

Let $U_1, U_2, U_3, U_4, U_5, U_6$ be the leftmost components of $V_1, V_2, V_3, V_4, V_5, V_6$ respectively. The input alphabet of the small topologies automaton will consist of the two disjoint lists. Namely the join irreducibles of the lattice under inclusion generated by U_1, U_2 and U_3 which consist of

$$U_1, U_2, U_3, U_1 \cap U_2, U_2 \cap U_3$$

and the set of join irreducibles of the lattice under inclusion generated by U_4, U_5 and U_6 which consists of

$$U_4, U_5, U_6, U_4 \cap U_5, U_5 \cap U_6.$$

In the notation of [24], the sets $V_i, i = 1, \ldots, 6$, correspond to an open cover of the graph of G' restricted to A. The sets of the cover are of the form $V_i = A_i \times B_i$, with $1 \leq i \leq 6$ where
$A_i = U_i \times \{on\}$ $1 \leq i \leq 3$,
$B_1 = \{on\}$, $B_2 = K$, $B_3 = \{off\}$; and
$A_i = U_i \times \{off\}$, $3 \leq i \leq 6$,
$B_4 = \{on\}$, $B_5 = K$, $B_6 = \{off\}$.
The finite automaton in the small topologies described in [24] assigns to each join-irreducible in the lattice generated by the open sets A_i, a set of control laws. That is, we attach to every non-empty join irreducible A_i' in the lattice generated by the A_i's, an open set

$$O(A_i') = \cup_{z \in \Gamma_i} B_z,$$

where $\Gamma_i = \{z \mid A_i' \subseteq A_z\}$.
In our case it is easy to check that we obtain the following assignments of $O(A_i')$ for the join-irreducibles A_i':

1. the sets $U_i \times \{on\}$, $1 \leq i \leq 3$ are mapped respectively to $\{on\}, K, \{off\}$.
2. the sets $U_i \times \{off\}$, $3 \leq i \leq 6$ are mapped respectively to $\{on\}, K, \{off\}$.
3. each of the following four join irreducibles, $(U_1 \cap U_2) \times \{on\}), (U_2 \cap U_3) \times$ $on\}$, $(U_4 \cap U_5) \times \{off\}, (U_5 \cap U_6) \times \{off\}$, is mapped to K.

Let $H(u, k)$ be any set-valued function which is consistent with the above assignments where u ranges over the set U of join irreducibles in the lattices generated by U_1, U_2, U_3 and by U_4, U_5, U_6. Formally, the finite automaton in the small topologies corresponding to the above data is the following:

1. The set of states $S = K$.
2. The input alphabet is the set U.
3. The output alphabet $V = K$.
4. The nondeterministic output function is based on the set-valued function H described in the assignments above.
5. The transition table $M : U \times K \to K$ is defined by $M(u, k) = H(u, k)$.

The automaton can be used for control as follows. Let y be a water level and k be the automaton current state.

1. The analog to digital converter transforms y into the least join-irreducible u that contains y.
2. The automaton maps u nondeterministically into a pump state $k' \in H(u,k)$, and outputs k' to the plant.

This automaton is parameterized by the ϵ entering the definitions of U_i'. Are there values of ϵ which guarantee that water level trajectories arising from the automaton satisfy the control requirements? While considering this question we may ask whether the automaton output function is related to a suitable feedback control function. Should it happen to be a guiding feedback control function for some region of Q, then the control automaton would satisfy the control requirements if it began its operation in that region.

Consider the following "feedback control function": $f(y,k) = H(u,k)$, where u is the least join-irreducible that contains y. It is then easy to see that:

When $k = on$:

$$f(y,k) = \begin{cases} K & \text{if } y \in (g-e-\epsilon, g+e+\epsilon) \\ on & \text{if } y \in [h-e, g-e-\epsilon] \\ off & \text{if } y \in [g+e+\epsilon, v-a\cdot d] \end{cases}$$

When $k = off$,

$$f(y,k) = \begin{cases} K & \text{if } y \in (h-e-\epsilon, h+e+\epsilon) \\ on & \text{if } y \in [u+b\cdot d, h-e-\epsilon] \\ off & \text{if } y \in [h+e+\epsilon, g+e] \end{cases}$$

We have three objects now: the finite automaton in the small topologies, the corresponding function f, and the control automaton associated with f. It is easy to see that each of the three objects have the same set of water level trajectories over the region A generated by the object. It follows that if f is a guiding feedback control function over A, then the finite automaton with small topologies is correct.

We can conclude that f is a guiding feedback control function from the following general fact and Theorem 17.

Proposition 18. *Suppose $A \subseteq PS \times K$ and f, F are two set-valued functions over A with values subsets of K. Suppose that the graph of f is a subset of the graph of F and F is a guiding feedback control function. Then so is f.*

Proof It is clear that all the plant trajectories generated by f constitute a subset of those generated by F. The conclusion desired is immediate. □

Consider F, which is determined by g, h and $e' = e+\epsilon$. Assume that the premises of Theorem 17 are satisfied by this data for some $\epsilon_0 > 0$. It follows from Theorem 17 that F is a guiding control function. It follows from the proposition above that so is f for any $\epsilon \leq \epsilon_0$.

Remark The control automaton $Aut1$ described above is a formal representation of the controller from [1]. That paper does not mention using a sampling

interval $\Delta > 0$. We can interpret this as meaning that water level is measured and tested continuously. Continuous measurement and testing in the presence of pump delay can cause the above control automaton and the controller from [1] to produce an infinite number of outputs in a finite interval of time, a physical impossibility. Consider a time t at which the automaton outputs a request to change the pump state. Suppose that just prior to that time the pump was "on" and the state of the automaton was *son*. Suppose that the pump delay is $d > 0$. Since the water level continues to increase during the delay, and the automaton continuously samples the input, the automaton senses the condition $y \geq g$ at all times in the interval $(t, t+d)$. Thus the automaton will produce an essential output at each time in that interval. Our assumption that we sample (measure, sense) after each interval of length $\Delta > d > 0$ eliminates this source of unrealizable behavior. Sampling at times separated by a positive bound $\Delta > 0$ cannot be dispensed in modeling a plant with delays.

Later papers will investigate open covers and the corresponding finite automata with small topologies for a variety of control problems.

References

1. R. Alur, C. Courcoubetis, T. Henzinger, Pei-Hsin Ho, Hybrid Automata: An Algorithmic Approach to the Specification and verification of hybrid Systems, Workshop on Hybrid Systems, Denmark, October 1992.
2. P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, eds., *Hybrid Systems II*, Lecture Notes in Computer Science vol. 999, Springer-Verlag, (1995).
3. K.R. Apt and E-R. Olderog, *Verification of Sequential and Concurrent Programs*, Springer-Verlag, 1991.
4. J. P. Aubin, *Differential Inclusions, Set Valued Maps, and Viability*, Springer-Verlag, 1984.
5. J. P. Aubin, *Set Valued Analysis*, Birkhauser, 1990.
6. J. P. Aubin, *Viability Theory*, Birkhauser, 1991.
7. J-P.Aubin and I. Ekeland, *Applied Nonlinear Analysis*, Wiley-Interscience, 1984.
8. L.D. Berkovitz, Thirty Years of Differential Games, in Emilio O. Roxin (editor), *Modern Optimal Control*, Marcel Dekker, Inc., 1989.
9. Robert S. Boyer, Milton W. Green, J. Strother Moore, The Use of a Formal Simulator to Verify a Simple Real Time Control Program,Technical Report No. ICSCA-CMP-29, Software Systems Science, National Science Foundation, Washington, D.C. 20550, July, 1982. DTIC Selected Sept. 16 1983.
10. J. R. Buchi, *The Collected Works of J. Richard Büchi* (S. MacLane,. Siefkes, eds.), Springer-Verlag, 1990. 1990.
11. K. M. Chandy and J. Misra, *An Introduction to Parallel Program Design*, Addison-Wesley, 1988.
12. A. F. Filippov, *Differential Equations with Discontinuous Right Hand Side*, Kluwer Academic Publishers, 1988.
13. A. Friedman, *Differential Games*, Wiley- Interscience, 1971.
14. X. Ge, W. Kohn, A. Nerode, and J.B. Remmel, "Algorithms for Chattering Approximations to Relaxed Optimal Control. MSI Tech. Report 95-1, Cornell University. (1995)
15. X. Ge, W. Kohn, A. Nerode, and J.B. Remmel, "Feedback Derivations: Near Optimal Controls for Hybrid Systems", *Hybrid Systems III*, Lecture Notes in Computer Science 1036, Springer-Verlag, (1995), 76-100.

16. R. Grossman, A. Nerode, H. Rischel, A. Ravn, eds., *Hybrid Systems*, Springer Lecture Notes in Computer Science 736, (1993).
17. Y.Gurevich and L. Harrington, Trees, Automata and Games, Proc. of the 14th Ann. ACM Symp. on Theory of Comp., pp. 60-65, 1982.
18. O. Hajek,*Pursuit Games*, Mathematics in Science and Engineering, vol. 120, Academic Press, New York, 1975.
19. J. Hilgert, K. H. Hofmann, and J. Lawson, *Lie Groups, Convex Cones, and Semigroups*, Oxford Clarendon Press, 1988.
20. R. Isaacs,*Differential Games*, SIAM Series in Applied Mathematics, John Wiley and Sons, Inc., 1965.
21. W. Kohn, Hierarchical Control Systems for Autonomous Space Robots, Proc. AIAA, 1988.
22. W. Kohn and A. Nerode, An Autonomous Control Theory: An Overview, Proc. IEEE CACSD92, Napa Valley, March, 1992.
23. W. Kohn and A. Nerode, Multiple Agent Autonomous Control Systems, Proc. 31st IEEE CDC Tucson, Ar., 2956-2966, 1993.
24. W. Kohn and A. Nerode, Models for Hybrid Systems: Automata, Topologies, Controllability, Observability, in [16], 1993.
25. W. Kohn and A. Nerode, Multiple Agent Autonomous Control, A Hybrid Systems Architecture, to appear in *Logical Methods: A Symposium in honor of Anil Nerode's 60th birthday*, Birkhauser, 1993.
26. W. Kohn, A. Nerode, and J.B. Remmel, " Hybrid Systems as Finsler Manifolds: Finite State Control as Approximation to Connections", In [2], (1995), 294-321
27. Kohn, W., Nerode, A. and Remmel, J.B., "Continualization: A Hybrid Systems Control Technique for Computing", Proceedings of CESA'96, (1996), 507-511.
28. Kohn, W., Nerode, A. and Remmel, J.B., "Feedback Derivations: Near Optimal Controls for Hybrid Systems", Proceedings of CESA'96, (1996), 517-521.
29. N.N. Krasovskii and A.I. Subbotin, *Game-Theoretical Control Problems*, Springer-Verlag, 1988.
30. Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems*, Springer-Verlag, 1992.
31. E. B. Lee and L. Marcus. Foundations of optimal control theory, John Wiley & Sons, 1967.
32. K. Marzullo, F. Schneider, N. Budhiraja, Derivation of Sequential, Real-Time, Process-Control Programs, Technical Report 91- 1217, Department of Computer Science, Cornell University, Ithaca, NY 14853-7501.
33. R. McNaughton, Infinite Games Played on Finite Graphs, Annals of Pure and Appl. Logic, 65 (1993), 149-184.
34. A. Nerode and J.B. Remmel, A Model for Hybrid Systems, Hybrid System Workshop Notes, MSI, Cornell University, Ithaca, NY, June 1990.
35. A. Nerode, J.B. Remmel and A. Yakhnis, "McNaughton Games and Extracting Strategies for Concurrent Programs", to appear Annals of Pure and Applied Logic.
36. A. Nerode, A. Yakhnis, Modelling Hybrid Systems as Games, Proceedings of the Conference on Decision and Control, pp.2947-2952, 1992.
37. A. Nerode, A. Yakhnis, Hybrid Games and Hybrid Systems, Technical Report No. 93-77, October 1993, Mathematical Sciences Institute, Cornell University, 409 College Ave., Ithaca, NY 14850.
38. A. Nerode, A. Yakhnis, V. Yakhnis, Concurrent Programs as Strategies in Games, in *Logic from Computer Science*, (Y. Moschovakis, ed.), Springer-Verlag, 1992.
39. A. Nerode, A. Yakhnis, V. Yakhnis, Distributed Programs as Strategies in Games, in a volume honoring Anil Nerode 60th birthday, Birkheuser, 1994.

40. L.S. Pontryagin, On the Theory of Differential Games, Russian Mathematical Surveys 21 (No.4), pp. 193-246, 1966.
41. J. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall 1991.
42. E. D. Sontag, *Mathematical Control Theory*, Springer-Verlag, 1990.
43. V. I. Utkin, *Sliding Modes in Control Optimization*, Springer-Verlag, 1992.
44. J. Warga, *Optimal Control of Differential and Functional Equations*, Academic Press, 1972.
45. J. Warga, Some Selected Problems of Optimal Control, in Emilio O. Roxin (ed.), *Modern Optimal Control*, Marcel Dekker, Inc., 1989.
46. A. Yakhnis, Game-Theoretic Semantics for Concurrent Programs and Their Specifications, Ph. D. Diss., Cornell University, 1990.
47. A. Yakhnis, Hybrid Games, Technical Report 92-38, Mathematical Sciences Institute, Cornell University, October 1992.
48. A. Yakhnis, V. Yakhnis, Extension of Gurevich-Harrington's Restricted Memory Determinacy Theorem, Ann. Pure and App. Logic 48, 277-297, 1990.
49. A. Yakhnis, V. Yakhnis, Gurevich-Harrington's games defined by finite automata, Ann. Pure and App. Logic 62, 265-294, 1993.